Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Silvo Gazvoda

Analiza delovanja računalniških omrežij s programskim orodjem CACTI

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

prof. dr. Miha Mraz
MENTOR

doc. dr. Miha Moškon
SOMENTOR

Ljubljana, 2014

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kandidat naj v diplomskem delu izvede analizo možnosti nadzora računalniških omrežij. Pri tem naj izvede tako pregled strategij in standardov nadzora, kot tudi pregled razpoložljivih orodij. V vzorčnem javno dostopnem orodju Cacti naj naredi analizo tipičnih primerov nadzora delovanja naprav.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.
S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom prof. dr. Mihe Mraza in somentorstvom doc. dr. Mihe Moškona,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Silvo Gazvoda, Ljubljana, september 2014.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Silvo Gazvoda

Analiza delovanja računalniških omrežij s programskim orodjem CACTI

POVZETEK

V diplomski nalogi smo opredelili tehnike in pristope, ki jih najpogosteje srečamo pri izvajanju nadzora in upravljanja omrežij. Izbira tehnike, s katero se lotimo nadzora omrežja, zavisi od kompleksnosti nadzorovalnih metrik in vzpostavljene infrastrukture. Pri nadzoru zmogljivosti smo spoznali arhitekturo, v kateri centralizirani upravljavec pošilja poizvedbe upravljanim napravam. Naprave svoje zmogljivostne parametre izpostavijo v objekte definirane znotraj baze upravljavskih informacij. Vrednosti parametrov se med upravljavcem in napravami najpogosteje prenašajo v podatkovnih enotah protokola SNMP (angl. Simple Network Management Protocol). Nadaljevali smo z analizo programskega orodja Cacti. Orodje omogoča opazovanje različnih metrik zmogljivosti omrežnih virov skozi množico grafov. V zadnjem delu smo se lotili vzpostavitve sistema za nadzor omrežja v katerem smo opazovali lokalne in oddaljene naprave. Orodje Cacti se je izkazalo kot prilagodljivo orodje za nadzor omrežij.

Ključne besede: Cacti, nadzor omrežij, protokol SNMP, MIB, orodje RRDTool, upravljani objekti

University of Ljubljana
Faculty of Computer and Information Science

Silvo Gazvoda

Computer networks analysis with Cacti

ABSTRACT

In this thesis, we have identified techniques and approaches that are most commonly encountered in network management systems. We have described availability and performance monitoring techniques. Selection of monitoring technique depends on the complexity of monitored parameters and preliminary established architecture. Network monitoring suggests architecture in which centralized manager collects and analyses data from managed devices. Managed devices expose their network statistics through objects defined within the Management Information Base. Simple Network Management Protocol allowing the manager to communicate with the managed hosts. Afterwards we analyse network monitoring and graphing tool Cacti. Cacti allows a manager to poll managed devices and graph the gathered data. In the last part we cover the details of establishing network management system in which we monitor local and remote network resources. Cacti has emerged as adaptable tool for monitoring computer networks.

Key words: Cacti, network monitoring, SNMP, MIB, RRDTool, managed objects

ZAHVALA

Rad bi se zahvalil mentorju prof. dr. Mihi Mrazu in somentorju doc. dr. Mihi Moškonu za strokovno pomoč in potrpljenje pri nastajanju diplomske naloge. Prav tako bi se rad zahvalil staršema, ki sta mi omogočila študij in me podpirala skozi vso študijsko pot. Zahvala gre tudi ostalim članom družine in prijateljem za vse pozitivne želje ter spodbude pri študiju in pisanju diplomske naloge.

— Silvo Gazvoda, Ljubljana, september 2014.

KAZALO

1 Uvod

V zadnjem času je opazna vse hitrejša rast uporabe spleta in posledično tudi širjenje računalniških omrežij. Kot rednemu uporabniku se mi venomer poraja vprašanje, kako ponudniki internetnih storitev zagotavljajo dosegljivost in zmogljivost ponujenih storitev. Uporabnikom je visok nivo dosegljivosti in zmogljivosti, ki ju zagotavlja ponudnik, zelo pomemben kriterij. Uporabniki namreč ob nepredvidenih dogodkih kot so visoka latenca, nedosegljivost storitev ali nižja prenosna hitrost do ponudnikov postanemo zelo kritični. Navedeni dogodki predstavljajo le nekaj primerov, ki jih ponudniki storitev in administratorji omrežij želijo povsem odpraviti ali pa vsaj znižati verjetnost njihovih porajanj. Pereč problem se do neke mere rešuje z nadzorom in upravljanjem omrežnih virov. Z doslednim nadziranjem stanja omrežja lahko identificiramo in odpravimo potencialne anomalije. Orodja, ki so na voljo skrbnikom in administratorjem omrežij, ponujajo zelo širok nabor metrik in parametrov, ki jih je v omrežju mogoče nadzorovati. Kateri zmogljivostni parameter nadzorovati in kakšne so dopustne vrednosti, ki jih lahko zavzame, je odvisno od razpoložljivosti in zmogljivosti virov v omrežju. Kot primer si oglejmo strežnik, ki v določenem delu dneva ni dostopen. Z nadzorom strežnika bi pridobili in-

formacije, ki bi nam služile pri identifikaciji in odpravi problema. Zavedati se moramo, da zgolj s spremljanjem in upravljanjem virov ne zagotavljamo najvišje dosegljivosti. Ob zaznavi nenadne odpovedi strojne opreme se nam brez vpeljane redundance niža odstotek razpoložljivosti. Z metodo redundance obstoječemu sistemu kot celoti dvignemo zanesljivost, toda sama redundanca ne odtehta prednosti, ki jih pridobimo z metodami in pristopi nadzora in upravljanja opazovanih sistemov.

Bogat nabor orodij za vzpostavitev nadzornih sistemov v omrežju sestavlja tudi veliko odprtokodnih rešitev. Eno izmed takih orodij je programsko orodje Cacti. Slednje omogoča vzpostavitev centralnega opazovalca, preko katerega skrbnik omrežja spremlja dosegljivost in stanje posameznih omrežnih virov. Izbrano orodje bomo v nadaljevanju razčlenili in ga podrobneje analizirali ter spoznali njegove funkcionalnosti skozi primere nadzora omrežnih naprav. V pričujočem delu se bomo najprej osredotočili na področje nadzora in upravljanja omrežij. Srečali se bomo z osnovnimi pojmi in parametri, s katerimi podamo zmogljivost opazovanega omrežja. Obravnavo področja bomo razčlenili na nadzor dosegljivosti in zmogljivosti. Seznanili se bomo z metodami in strukturo parametrov, ki jih bodisi opazujemo ali spreminjamo. Drugo poglavje bomo zaključili s podrobnejšim pregledom komunikacijskega protokola SNMP (angl. *Simple Network Management Protocol*) in kratko analizo dveh orodji. V tretjem poglavju bomo temeljitejše analizirali orodje Cacti. Obravnava vključuje metodi shranjevanja in analiziranja podatkov, po katerih orodje pošilja poizvedbe opazovanim omrežnim napravam. Jedro obravnavanega orodja tvori orodje RRDTool, ki poleg shranjevanja omogoča tudi grafično predstavitev zmogljivostnih podatkov. Funkcionalnosti orodja bomo spoznali tudi skozi spletni vmesnik, preko katerega dostopamo do množice grafov, ki odražajo stanje nadzorovanih naprav. V zaključni fazi diplomskega dela se z orodjem Cacti lotimo opazovanja osebnega računalnika in usmerjevalnika v lokalnem omrežju. Praktični del bomo zaključili z vzpostavitvijo nadzora oddaljenega računalnika z varnostnimi mehanizmi, ki jih nudi tretja verzija protokola SNMP. Pri vseh primerih bomo opisali korake konfiguranja orodja in opazovanih naprav.

Pri obravnavi področja smo izhajali iz vira [1]. V navedenem viru najdemo obsežnejši pregled nadzora kot tudi upravljanja omrežij. Naslednji vidnejši vir predstavlja Ciscov priročnik [2], v katerem zasledimo različne pristope in tehnike ter orodja, ki se jih poslužujejo skrbniki omrežij. Pri analizi orodja Cacti pa smo si pomagali s priročnikom [3].

2 Opis problema

V nadaljevanju bomo spoznali osnovne pojme, ki jih srečamo na področju nadzora in upravljanja računalniških omrežij. Posvetili se bomo obravnavi področja in spoznali različne metode nadzora in upravljanja. Seznanili se bomo tudi s komunikacijskimi protokoli in strukturo izmenjanih podatkov.

2.1 Osnovni pojmi upravljanja in nadzorovanja omrežij

Vse hitrejši razvoj tehnologije vpliva tudi na rast računalniških omrežij. Zaradi vse večje omrežne infrastrukture se poraja zahteva po nadzoru in upravljanju virov (resursov), ki sestavljajo omrežje. Z vidika upravljanja nas pogosto zanima katere vire vsebuje infrastruktura, koliko prometa gre skozi posamezne vmesnike, katerega prometa je največ in kdaj. Pod pojmom upravljanje omrežja razumemo širok nabor metod, aktivnosti, postopkov in rabo orodij za zanesljivo vzdrževanje in administriranje strojne in programske opreme omrežnih virov. Upravljanje je pomembno tudi pri odpovedih in napakah v omrežju. Stalno spremljanje omrežja nam omogoča, da hitreje najdemo in odpravimo točko odpovedi oziroma napake. Potrebo po upravljanju omrežja nam narekuje tudi

zgodovina. V prvih letih interneta takrat imenovanega ARPANET ni bilo govora o mehanizmih za spremljanje prometa ali nadzorovanja delovanja komponent v omrežni infrastrukturi. Oktobra 1980 se je zgodila večja odpoved ARPANET-a, ki je dokumentirana v RFC-ju 789 [4]. Napaka je povzročila popolno nedosegljivost omrežja za nekaj ur. RFC 789 je študij primera specifičnega problema, ki lahko nastane pri večjih porazdeljenih omrežjih. V omenjenem RFC-ju zasledimo sklep, da bi z vgrajenim programskim alarmnim sistemom lahko hitreje vzpostavili omrežje ali občutno skrajšali čas odpovedi.

Administrator sistema, v katerem je vzpostavljeno orodje za upravljanje in nadzor omrežja, se sooča z upravljanjem različnih aktivnosti. Administrator stalno spremlja dosegljivost omrežnih virov. V primeru zaznane napake se lahko tako odzove hitreje, torej še preden je napaka javljena s strani končnega uporabnika. Naslednja aktivnost, pri kateri ima administrator korist, je spremljanje prometa za namene postavitve oziroma prilagoditve omrežnih virov. Pri spremljanju prometa opazimo različne vzorce. Ti vzorci povejo ali bi s premestitvijo kosa strojne opreme v drug segment podomrežja zmanjšali obremenitev. Torej lahko zgolj s spremljanjem omrežnega prometa vplivamo na povečanje zmogljivosti brez nakupa in vpeljave nove opreme. Zaznavanje vdorov je naslednja aktivnost, ki zahteva administratorjevo obveščenost. V primeru, da promet prihaja iz sumljivega izvora ali je naslovljen na naslov, ki lahko predstavlja nevarnost, se administrator ustrezno odzove. Ravno tako želimo preprečiti ali omejiti promet, ki vsebuje karakteristike prometa, katerega smo že prepoznali, da je nevaren. Spremljanje omrežja oz. spremljanje parametrov storitve predstavlja pomembno aktivnost za zagotavljanje podpore na podlagi dogovora o ravni storitve (angl. *service level agreement*, SLA). Dogovor o ravni storitve vsebuje opis storitev, metrike in njihove sprejemljive vrednosti, katere ponudnik omrežja zagotavlja strankam [1]. Sporazum navaja:

- omrežne metrike, kot so izguba paketov, latenca, trepetanje (angl. *jitter*) oziroma variabilnost zakasnitev paketov,
- pričakovanja in odgovornosti sodelujočih strani,
- predvideni minimalni čas prejema obvestila o načrtovanem vzdrževanju,
- čas za odziv in razrešitev nastalih problemov.

Ključni del upravljanja omrežja je spremljanje njegovega delovanja (angl. *Network Monitoring*). Spremljanje omrežij je široko področje, ki zajema aktivnosti merjenja,

primerjanja in opazovanja delovanja računalniškega omrežja. Celoten sistem, ki ga vzpostavimo za spremljanje delovanja omrežja, bomo poimenovali sistem spremljanja omrežja (angl. *Network Monitoring System*). S sistemom spremljanja omrežja opazujemo delovanje notranjega omrežja ter zaznavamo oziroma odkrivamo napake. S spremljanjem omrežja poskušamo zaznati in preprečiti potencialne napake in izpade v omrežju. Sistem za nadzor omrežja zgradimo s programsko opremo oziroma s kombinacijo strojne in programske opreme. Proces spremljanja delovanja naprav lahko izvajamo na različnih operacijskih sistemih z množico funkcij, ki obsegajo spremljanje strežnikov, stikal, mobilnih telefonov in usmerjevalnikov. Naprave omrežja spremljamo po principu izpraševanja. Sistem pošilja poizvedbo v nadzorovana vozlišča ob rednih časovnih intervalih. Rezultati opravljenih izpraševanj podajo stanje omrežnih naprav. Na podlagi rezultatov lahko določimo vrednosti zmogljivostnih metrik, dejavnosti in v primeru napak generiramo signale, ki opozarjajo na napako v omrežju. Prejeti rezultati povpraševanja se shranjujejo za namene poročanja in spremljanja v daljšem časovnem obdobju.

V povezavi s področjem spremljanja omrežja in s tem posledično tudi zagotavljanjem bolj robustnega sistema, zasledimo izraze kot so pet devetic, MTTR in ostali. Vrednost MTTR (angl. *Mean Time to Repair*) označuje povprečni čas izpada storitve ali naprave. Čas je merjen od trenutka izpada vse do trenutka povrnitve v popolno delujoče stanje. Razpoložljivost (angl. *avaliability*) poda odstotek časa dosegljivosti sistema ali storitve v podanem časovnem obdobju. V obdobju 90 dni 90% razpoložljivost pomeni, da je naprava aktivna in delujoča 81 dni. Ostalih 9 dni je naprava neaktivna in predstavlja parameter MTTR, ki v tem zgledu znaša 9 dni. Obdobje v katerem je naprava aktivna predstavlja parameter MTBF; povprečni čas med odpovedima (angl. *Mean Time Between Failure*). Slednji podaja povprečni čas dosegljivosti oziroma delovanja naprave, sistema ali dela sistema pred naslednjo odpovedjo. Merjen je od trenutka začetka popolnega delovanja do trenutka odpovedi. Izraz pet devetic izraža sistem, katerega razpoložljivost znaša 99,999%. Po viru [2] je izraz najbolj prepoznaven in uporabljen v krogih upravljavcev in vodstva kadar teče razprava o zmogljivosti omrežja. Sistem z zanesljivostjo petih devetic v obdobju enega leta dopušča čas izpada, ki traja le 5 minut in 15,36 sekund, upoštevajoč izvajanje vseh predvidenih vzdrževanj in odpravo nastalih napak oziroma nepredvidenih dogodkov.

V nadaljevanju si bomo ogledali tehnike oziroma metode nadzorovanja omrežja. Začeli bomo z enostavnejšimi in nadaljevali z naprednejšimi metodami, kot je nadzor zmoglji-

vosti. Delitev temelji na kompleksnosti infrastrukture omrežja, ki ga upravljamo. Ravno tako bomo spoznali komponente, ki sestavljajo infrastrukturo ter komunikacijske protokole. Protokol SNMP (angl. *Simple Network Management Protocol*) igra osrednjo vlogo pri nadzoru ter upravljanju omrežja in omogoča komunikacijo med upravljavcem in nadzorovanimi napravami. V naslednjih poglavjih bomo predstavili tudi strukturo podatkov, ki se prenašajo med entitetami.

2.2 Nadzor dosegljivosti

Nadzor dosegljivosti naprav v omrežju je preprosta in najosnovnejša oblika nadzora omrežja. Pri tem načinu preverimo ali je naprava dosegljiva. Metoda temelji na pošiljanju paketa ICMP (angl. *Internet Control Message Protocol*) ciljni napravi oz. gostitelju z ukazom `ping` [5]. Uspešno prejetje odgovora potrjuje dosegljivost gostitelja. Večina naprav podpira `ping` ne glede na operacijski sistem, zato lahko obravnavani način preverjanja dosegljivosti uporabimo na večini omrežnih naprav. Prejem pozitivnega odgovora potrjuje tudi dejstvo, da na ciljni napravi sklad TCP/IP deluje pravilno. To metodo smo opredelili kot enostavnejšo, saj izključuje potrebo po namestitvi programske opreme oziroma agenta na nadzorovano vozlišče. Ukaz `ping` poleg obhodnega časa (angl. *Round trip time*), ki je podan v milisekundah, beleži tudi izgube paketov. Po koncu postopka prejmemo navedene informacije, ki odražajo stanje in razmere v omrežju. S primerjavo novo prejetega časa obhoda in preteklimi vrednostmi pripravimo poročila, ki nudijo administratorju omrežja celovit pogled nad dostopnostjo končnih naprav in identifikacijo ozkih grl, ob izpadu komunikacije pa administrator prejme ustrezno opozorilo. Poleg vseh prednosti, kot so enostavnost implementacije in možnost nadzorovanja večine omrežnih naprav, obstajajo tudi pomanjkljivosti. Metoda ne omogoča nadzora storitev, kot so HTTP, IMAP ali SMTP. Tehnika je omejena le na nadzor naprav na omrežnem sloju (angl. *network layer*). V omrežjih, kjer se z redundantnimi vmesniki ali napravami zagotavlja večja zanesljivost, lahko obravnavana tehnika proizvede napačne rezultate nadzora, saj redundantne komponente v stanju pripravljenosti obravnava kot nedosegljive, ali v stanju odpovedi. Pri preverjanju dosegljivosti s pričujočo tehniko se generira ICMP promet, ki je pri večini omrežij blokiran s strani požarnega zida. V tem primeru je preverjanje dosegljivosti neuspešno. Naslednja pomanjkljivost, na katero lahko naletimo pri generiranju ICMP prometa, je delovanje usmerjevalnikov ali ostalih omrežnih naprav v

primeru visoke obremenitve. V teh primerih je obremenjenost oz. izkoriščenost CPE na opazovani napravi visoka in lahko se zgodi, da odgovora na zahtevo ICMP ne prejmemo, čeprav naprava nemoteno deluje. Razlog takega obnašanja je dodelitev nižje strežne prioritete prometu ICMP. Naslednji faktor, ki ga je potrebno upoštevati je dejstvo, da nekateri ponudniki internetnih storitev blokirajo promet ICMP. Vsi našteti faktorji lahko vplivajo na obhodni čas in na ostale parametre pri izvajanju nadzora.

Pri tehniki `ping` za potrebe nadzora sicer ni bilo potrebe po namestitvi programskih orodij, storitev ali protokolov na nadzorovanih napravah. Potrebne storitve, protokoli, demoni in programi so vključeni v predhodno nameščene operacijske sisteme in gonilnike. Tehnika je neinvazivna, saj na merjencu ni potrebno namestiti posebne programske opreme, ki pri svojem delovanju zaseda razpoložljive vire (CPE, pomnilnik itd.) merjenca. Na eni strani rokujemo z enostavno metodo, na drugi strani pa smo omejeni s količino oz. globino podatkov, ki jih na ta način pridobimo. Družino, kateri metoda pripada, imenujemo tudi *nadzorovanje brez agentov*. Izraz agent na področju nadzorovanja predstavlja program, ki teče na opazovani napravi, katerega namen je zbiranje in posredovanje podatkov preko omrežja na centralno točko. Informacijo se preko omrežja pošilja v standardiziranem formatu, kot ga definira protokol SNMP, ki ga bomo podrobneje spoznali v prihajajočih razdelkih. Naprednejša metoda, ki pripada družini *nadzorovanje z agenti*, vključuje tudi konfiguracijo opazovanih naprav (invazivno nameščanje agentov na napravah, s katerih zbiramo podatke). Z namestitvijo programske opreme na napravah pridobimo več in podrobnejše podatke. Z namestitvijo agentov na končnih točkah pridobimo bolj robusten sistem nadzora. Poleg tega nam interakcija med predhodno nameščenimi agenti omogoča tudi izvajanje operacij upravljanja sodelujočih naprav v postavljeni infrastrukturi. Skladno s pričakovanji so rešitve nadzora in upravljanja z agenti cenovno dražje od rešitev brez nameščenih agentov. Na tem področju obstajajo tudi odprtokodne rešitve. Ena izmed njih je programsko orodje CACTI, ki sodi v metodo nadzora zmogljivosti. Orodje vključuje komunikacijski protokol SNMP in sodelovanje agentov na končnih napravah. V naslednjem razdelku sledi vpogled v metodo nadzora, v katero sodi obravnavano orodje, nato pa še pregled protokola SNMP.

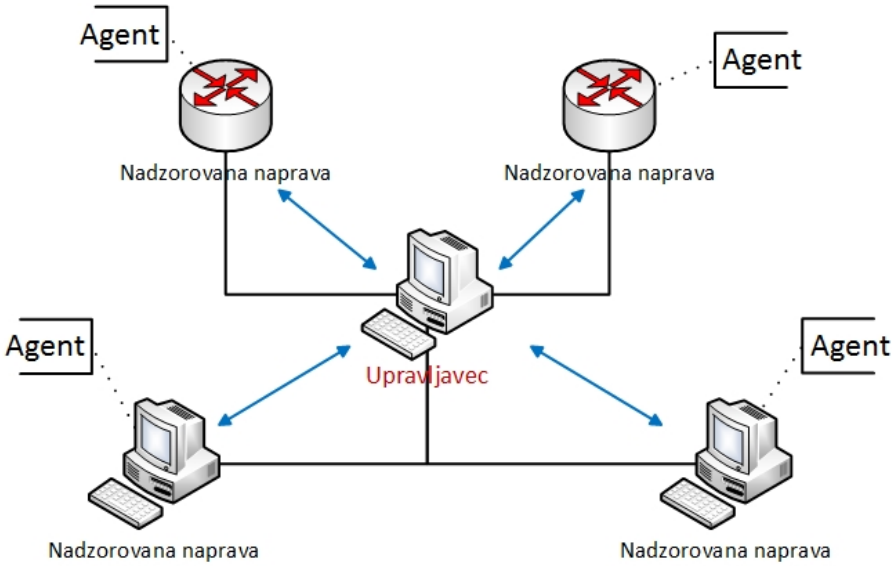
2.3 Nadzor zmogljivosti

V tem razdelku bomo obravnavali tehniko, ki kot sestavni del infrastrukture vključuje tudi nameščene agente na opazovanih končnih točkah. Agenti na nadzorovanih napravah posredujejo informacije centralni točki. Centralna točka v tem primeru predstavlja računalnik oz. strežnik, na katerem se vrši proces nadzora ostalih naprav. V proces nadzora je vključeno tudi izvajanje analize pridobljenih informacij in beleženje omrežne statistike. Najpogosteje nadzorovani parametri oz. metrike so po [2] naslednje:

- latenca (angl. *latency*),
- trepetanje (angl. *jitter*),
- propustnost (angl. *throughput*),
- izguba paketov (angl. *packet loss*),
- izkoriščenost CPE in pomnilnika.

Nadzorni sistem običajno uporablja SNMP protokol za komunikacijo z nadzorovanimi napravami. Sistem v enakih časovnih intervalih pošilja poizvedbe napravam in zbira informacije zmogljivostnih parametrov. Prejete informacije se na centralni točki shranijo v podatkovno bazo, katere lahko administrator uporabi za namene poročil, izvajanje analize in napovedovanje bremen. Časovni intervali v katerih se pridobivajo SNMP podatki oz. statistika običajno trajajo 5 minut [6]. Pri določanju dolžine intervala se upošteva velikost omrežja. Interval ne sme biti prekratek, saj bi s tem obremenili omrežne entitete. Interval mora biti dovolj dolg, da lahko v času njegovega trajanju pošljemo in pridobimo rezultate poizvedb iz celotnega omrežja. Pridobljena statistika ustreza povprečni aktivnosti omrežnih naprav in povezav v trajanju intervala. Nadzor predstavlja le del upravljanja omrežja, zato lahko infrastrukturo nadzorovanega omrežja predstavimo s komponentami upravljanega omrežja. Infrastruktura za upravljanje je prikazana na sliki 2.1.

Komponento upravljavca predstavljata človek in aplikacija, ki teče na centralizirani točki, s katere se upravlja oz. nadzoruje celotno omrežje. Nadzorovana naprava omrežno statistiko izpostavi v upravljane objekte (angl. *Managed objects*). Nadzorovana naprava predstavlja omrežne vire, kot so strežniki, usmerjevalniki, končne postaje (PC-ji, delovne

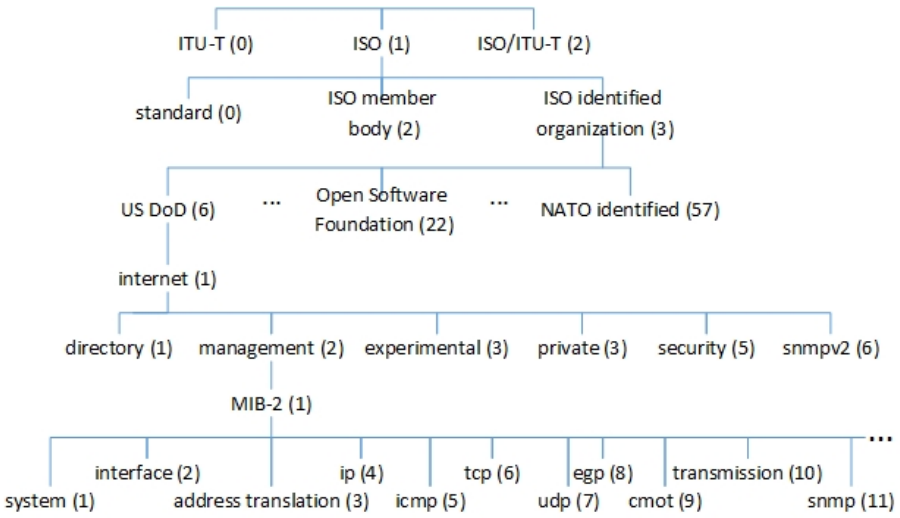


Slika 2.1 Komponente arhitekture upravljanja omrežja. Modre puščice predstavljajo komunikacijo med upravljavcem in nadzorovanimi napravami preko protokola SNMP.

postaje) itd. Upravljani objekti, ki so predstavljeni kot zbirka nadzorovanih parametrov, so zbrani v bazi upravljavskih informacij (angl. *Management Information Base*, MIB). Na vsaki nadzorovani napravi teče agent, ki komunicira z upravljavcem. Protokol za upravljanje omogoča upravljavcu pošiljanje poizvedb upravljanim napravam. Ravno tako lahko agenti brez predhodno poslane poizvedbe upravljavca obvestijo o izjemnih in nenavadnih dogodkih. SNMP je protokol aplikacijskega sloja in je tudi sestavni del IETF (angl. *Internet Engineering Task Force*) standarda, ki definira celotno zbirko tehnologij za upravljanje TCP/IP omrežja [7]. Ogrodje, ki ga definira standard, naslavlja definicijo baze upravljavskih informacij, jezik za definicijo objektov v bazi upravljalskih informacij, protokol SNMP in varnost. Poglejmo si posamezno komponento nekoliko podrobneje.

2.3.1 Baza upravljavskih informacij

Upravljana naprava vsebuje zbirko upravljanih objektov, v kateri vrednosti objektov odražajo trenutno stanje naprave. Te vrednosti agent centralnemu upravljavcu pošlje s SNMP sporočilom, kot odgovor na poizvedbo ali pa v obliki sporočila ob izjemnem dogodku v omrežju. Objekti so v MIB definirani z jezikom SMI (angl. *Structure of Management Information*), ki temelji na Abstract Syntax Notation One (ASN.1) [8]. Objekti so združeni v tako imenovane MIB module. Za standardizacijo MIB modulov



Slika 2.2 ASN.1 drevo identifikatorjev objektov [1].

omrežne opreme skrbi telo IETF. Zaradi velikega števila standardiziranih modulov in potrebe po identifikaciji posameznega objekta v modulu je IETF sprejel standardizirano poimenovanje organizacije ISO, ki predstavlja podmnožico standarda ASN.1. Poimenovanje objektov je hierarhično urejeno z drevesom identifikatorjev. Vsaka točka v drevesu je identificirana z zaporedjem imen ali številk, ki določajo pot od korena do točke v drevesu. Na vrhu drevesa najdemo organizacije za standardizacijo, pod določenim zaporedjem pa najdemo definicijo standardiziranih MIB modulov. Na sliki 2.2 je prikazano pričujoče drevo objektov. Oglejmo si kot primer zaporedje 1.3.6.1.2.1.11, ki predstavlja definicijo modula protokola SNMP, ki vsebuje informacije o delovanju protokola na obravnavanem sistemu. Če slednjemu zaporedju dodamo niz ≫.2≪ bo predstavljal parameter snmpOutPkts, ki vsebuje število SNMP sporočil posredovanih s strani SNMP protokola. Za definicijo objektov in parametrov se uporablja jezik SMI, ki ga bomo obdelali v naslednjem podpoglavju.

2.3.2 Jezik SMI

Jezik SMI je prilagojena podmnožica ASN.1, ki se uporablja za definicijo upravljanih objektov. Po [9] je obravnavani jezik razdeljen na tri dele: definicijo objektov, definicijo modulov in definicijo obvestil. Pri definiciji objekta se uporabi makro `OBJECT-TYPE`, ki specificira podatkovni tip, status in pomen definiranega objekta. Slednji konstrukt zahteva prisotnost štirih sklopov. Sklop `SYNTAX` določa podatkovni tip definiranega

objekta in lahko zavzame enega izmed 11 osnovnih podatkovnih tipov, ki so definirani v RFC 2578: INTEGER, Integer32, Unsigned32, OCTET STRING, OBJECT IDENTIFIER (poimenovanje objektov), IPaddress, Counter32, Counter64, Gauge32 (nenegativno število), TimeTicks (pretečeni čas od dogodka izražen v 1/100 sekunde) in Opaque. Sklop `MAX-ACCESS` določa ali lahko objekt beremo, pišemo, ga ustvarimo in ali je vrednost objekta vključena v obvestilo (accessible-for-notify). Sklop `STATUS` navaja ali je definicija objekta veljavna, zastarela, ali opuščena. Sklop `DESCRIPTION` navaja človeku prijaznejšo razlago definiranega objekta. Kot primer si oglejmo definicijo objekta `ipInDiscards`, ki je definiran v RFC4293 [10]. Zadnja vrstica kode 2.1 predstavlja ime objekta. Če navedeni primer izrazimo z zaporedjem številčnih identifikatorjev, bi mu ustrezal zapis 1.3.6.1.2.1.4.8.

Koda 2.1 Objekt `ipInDiscards` definira števec, ki beleži število zavrženih IP datagramov. Z vpeljavo definicije objekta `ipSystemStatsInDiscards` je postalo stanje pričujočega objekta opuščeno (angl. *deprecated*).

```
ipInDiscards OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
            "The number of input IPv4 datagrams for which no problems
             were encountered to prevent their continued processing, but
             which were discarded (e.g., for lack of buffer space).  Note
             that this counter does not include any datagrams discarded
             while awaiting re-assembly.

             This object has been deprecated, as a new IP version-neutral
             table has been added.  It is loosely replaced by
             ipSystemStatsInDiscards."
    ::= { ip 8 }
```

Pri povezovanju posameznih objektov v MIB module se uporabi makro `MODULE-IDENTITNY`. Navedeni primer objekta `ipInDiscards` je vključen v definicijo MIB modula za protokol IP, ki je specificiran v RFC 4293. Poleg protokola IP navedimo še MIB modula za TCP in UDP, ki sta zapisana v RFC4022 [11] in RFC4133 [12]. Vsi moduli poleg definicij objektov vsebujejo tudi kontaktne informacije avtorjev modula, zgodovino revizij, datum zadnje spremembe in tekstovni opis modula. S konstruktom `NOTIFICATION-TYPE` definiramo informacijo v obvestilih, katera so poslana ob izrednih dogodkih. Poslano sporočilo vsebuje opis, kdaj se obvestilo pošlje ter seznam vključenih vrednosti.

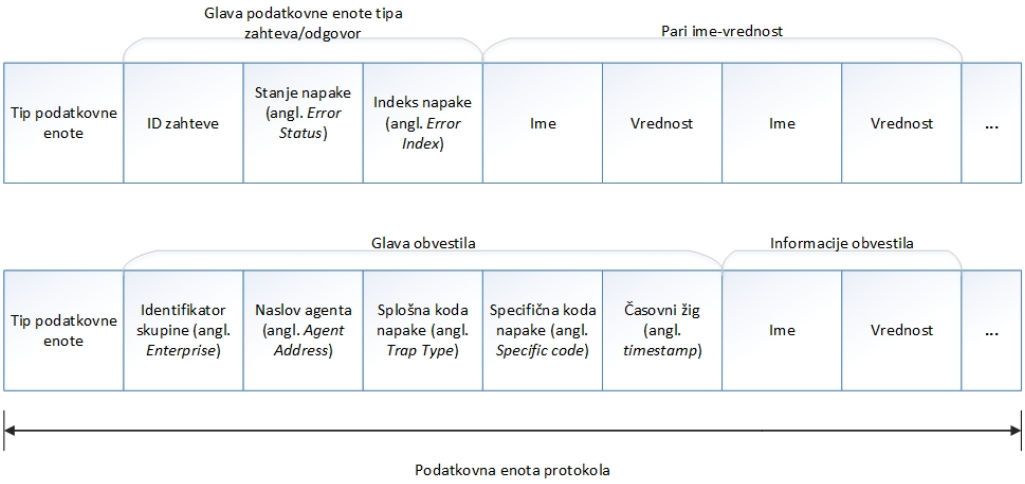
2.3.3 SNMP protokol

Protokol skrbi za prenos nadzorovanih podatkov, ki so predstavljeni kot baza upravljavskih informacij. SNMP se je razvil v treh verzijah. Poleg novih funkcionalnosti so se razvijali tudi varnostni mehanizmi, ki so bili predhodno predmet kritike. SNMP je protokol aplikacijske plasti in uporablja transportni protokol UDP [13]. Deluje v dveh načinih: zahteva-odgovor (angl. *request-response*) in obvestilo (angl. *trap message*). Pri prvem načinu upravljavska entiteta pošlje zahtevo nadzorovani enoti, na kateri SNMP agent prejeto zahtevo izvede in nanjo odgovori. Prejeta zahteva lahko predstavlja poizvedbo po upravljavskih podatkih ali nastavitev vrednosti v MIB objektu. Pri drugem načinu delovanja nadzorovana enota upravljavcu pošlje obvestilo o dogodku. Agent zahtevo prejme na UDP vratih 161, upravljavci pa poslušajo na vratih 162, na katera so običajno naslovljena SNMP obvestila (angl. *trap*). Navedena vrata se uporabljajo privzeto, vendar nekateri produkti omogočajo spremembo le-teh. V tem primeru je potrebno agente in upravljavsko entiteto primerno nastaviti. SNMP operacije so posredovane s sporočili. Za vsako operacijo je definirana podatkovna enota protokola (angl. *Protocol Data Unit*). Različice podatkovne enote protokola so prikazane na sliki 2.3. V RFC 3416 [14] je definiranih sedem tipov podatkovnih enot, ki jih agent in upravljavec uporabljata za medsebojno komunikacijo.

Podatkovne enote *GetRequest*, *GetNextRequest* in *GetBulkRequest* se pošiljajo v smeri od upravljavca proti agentu. Naštete enote predstavljajo poizvedbo po vrednostih enega ali več MIB objektov z naprave, na kateri prebiva agent. Po katerih vrednostih je bila zahteva poslana, je navedeno v posebnem odseku podatkovne enote imenovanem *Variable Bindings*, ki vsebuje pare ime-vrednost, s katerimi se označuje definirane objekte in njim pripadajoče vrednosti. Podatkovne enote protokola se med seboj razlikujejo po obsegu zahtevanih podatkov. S sporočilom tipa *GetRequest* lahko poizvedujemo po poljubnem nizu MIB vrednosti. Sporočilo tipa *GetNextRequest* se uporablja za branje vrednosti v tabeli ali seznamu MIB objektov. Zahteva slednjega tipa je uporabna predvsem pri zajemu podatkov ene ali več vrstic v dinamični tabeli. Kadar želimo prenesti večji segment tabele naenkrat, to storimo s pošiljanjem sporočila tipa *GetBulkRequest*. V tem primeru z eno zahtevo dostopamo do večje količine podatkov brez vnovičnega pošiljanja sporočila *GetNext*. Več MIB objektov lahko zahtevamo tudi z operacijo *Get*, ki jo pošljemo s sporočilom *GetRequest*, vendar so velikosti sporočila pogojena z zmo-

gljivostmi agenta. Agent po prejetju zahteve odgovori s sporočilom tipa *Response*, ki vsebuje identifikatorje zahtevanih objektov in njim pripadajoče vrednosti. Kadar želimo spremeniti vrednost objekta uporabimo operacijo *Set*. Upravljavec ima možnost spreminjanja vrednosti večjemu številu objektov hkrati. Vrednosti lahko spreminjamo le tistim objektov, ki imajo to omogočeno v MIB, ravno tako pa mora imeti upravljavec pisalni dostop do objektov na agentu. Ob uspešni nastavitvi objekta agent odgovori s sporočilom *Response* s kodo napake 0, ki označuje, da pri nastavitvi do napake ni prišlo. Z drugo verzijo protokola SNMP je prišel tudi mehanizem za komunikacijo med upravljavci. Za ta namen se uporablja sporočilo tipa *InformRequest*, s katerim se posreduje vrednosti MIB objektov. Prejemnik se na uspešen prejem sporočila odzove s pošiljanjem sporočila *Response* s kodo napake enako 0. Opisani tipi sporočil predstavljajo bodisi zahtevo bodisi odgovor. Naslednji tip sporočil so obvestila, ki predstavljajo način, s katerim agenti obvestijo upravljavce o določenem dogodku. Po prejemu obvestila se agentu potrditve ne pošlje. Upravljavec prejeto obvestilo pravilno interpretira le v primeru, da razume informacijo, ki jo nosi obvestilo. Obvestilo se najprej identificira na podlagi splošne številke pasti. Definiranih je sedem pasti (0-6). Prvih šest označuje dogodke kot so ponovni zagon agenta, ponovna inicializacija agenta, odpoved ali zagon vmesnika in nepooblaščeni dostop. Število 6 označuje past, ki je lastna proizvajalcem opreme ali uporabnikom, ki ne sodijo med ostale splošno definirane pasti. Pasti proizvajalcev so podrobneje identificirane s specifičnimi identifikatorji, kateri so uvrščeni pod vejo *enterprises* v MIB drevesu. Obvestila, ki jih je definiral Cisco v MIB drevesu, tako predstavimo z zaporedjem *iso.org.dod.internet.private.enterprises.cisco* ali s številčnimi identifikatorji 1.3.6.1.4.1.9. Pri splošnih obvestilih gre za prenos vrednosti objektov, ki so vgrajeni oz. vsebovani tako v agentu kot v upravljavcu. Pri specifičnih obvestilih pa je vsebina obvestila odvisna od proizvajalca.

Prvi verziji protokola predstavita pojem skupnosti (angl. *Community*) za vzpostavitev zaupanja med agentom in upravljavcem [13]. Pogoj za uspešno komuniciranje med agentom in upravljavcem je nastavitev vsaj enega imena skupnosti. Ime skupnosti igra vlogo gesla in omejuje aktivnosti, ki se izvajajo med komunikacijo na bralne aktivnosti, bralno-pisalne aktivnosti in prejem obvestil. Kadar SNMP upravljavec pošlje zahtevo agentu, se ime skupnosti upravljavca in ime skupnosti agenta primerjata. Če se niza ujemata, se upravljavec uspešno avtenticira in agent odgovori na zahtevo. Ime skupnosti se pošilja kot odkrito besedilo, zato je priporočljivo definirati obvestila, ki se pošljejo ob



Slika 2.3 Slika prikazuje podatkovni enoti protokola SNMP [1]. Zgornji del slike predstavlja podatkovno enoto sporočila tipa zahteva-odgovor. Podatkovna enota obvestila je prikazana na spodnjem delu slike.

vsakem neuspešnem poskusu avtentikacije. Največja težavo SNMPv1 in SNMPv2 predstavlja varnost. Varnost je vprašljiva pri prenosu imena skupnosti. Agent in upravljavec enkripcije namreč ne izvajata. Vsakdo v omrežju, ki prisluškuje prometu, lahko prestreže ime skupnosti in s tem pridobi nadzor nad nadzorovanimi napravami. Tveganje nekoliko zmanjšamo s požarnim zidom, s katerim omejimo prejem SNMP zahtev le s poznanih računalnikov. Zgolj z omejitvijo naslovov, s katerih naprava prejemlje SNMP zahteve, ne dosežemo popolne varnosti. Večjo varnost zagotovimo, če SNMP promet ni viden niti zunaj omrežja niti v določenih delih lokalnega omrežja. Ukrep zahteva ustrezno nastavitev požarnega zidu in usmerjevalnikov, s katerimi blokiramo neželene SNMP pakete. Če uporabljamo SNMPv1 ali SNMPv2 in želimo nadzorovati omrežje od doma, se moramo zavedati nižje zasebnosti SNMP prometa, ki potuje po internetu. Če načrtujemo nadzor in upravljanje oddaljenega omrežja, višjo zasebnost dosežemo z vzpostavitvijo navideznega zasebnega omrežja (angl. *Virtual Private Network*, VPN) ali drugo obliko tuneliranja prometa. V tem primeru boljšo rešitev predstavlja uporaba spletnega vmesnika, ki z integracijo strežniških rešitev omogoča tudi druge varnostne mehanizme kot je SSL. Pomanjkljivost varnosti prvih dveh verzij protokolov naslavlja SNMPv3. Tretja verzija protokola poleg avtentikacije na podlagi imen skupnosti, ki se uporablja pri verziji 1 in 2, definira tudi avtentikacijo na osnovi uporabniških imen. Pri avtentikaciji uporabnikov se uporabljajo MD5 [15] ali SHA [16] algoritmi, ki preprečujejo nešifrirano

pošiljanje imena skupnosti. Celotno SNMP sporočilo se kriptira (šifrira) z DES ali AES algoritmom. SNMPv3 definira tudi nadzor dostopa, s katerim omejimo pravice dostopa do MIB objektov. Uporabniku določimo, do katerih objektov ima dostop ter katere operacije lahko izvaja nad temi objekti.

2.4 Pregled orodij

V pričujočem poglavju bomo spoznali le nekaj orodij za nadzor in upravljanje omrežij. Administratorji omrežij imajo na voljo širok nabor rešitev, med katerimi se odločajo na podlagi zahtev in izzivov, ki jih bo orodje reševalo. Različna omrežja imajo različne potrebe nadzora in upravljanja. Pri manjših omrežjih iščemo naprimer ugodne rešitve z enostavno vzpostavitvijo in uporabo ter s čim širšim spektrom funkcionalnosti. Na voljo je veliko odprtokodnih rešitev, ki so prilagodljive v smislu obsega nadzora in upravljanja ter načina prikaza pridobljenih informacij in obvestil. Na voljo je veliko odprtokodnih produktov, saj je področja nadzora in upravljanja omrežja zelo široko. Poleg tega je nemogoče razviti orodje, ki bi pokrilo vsa področja. Nestandardne pristope opazimo tudi pri komercialnih rešitvah. SNMP protokol, ki smo ga obravnavali v prejšnjih razdelkih, je le en način komunikacije med omrežnimi entitetami. Preden se bomo lotili podrobne obravnave orodja CACTI, bomo grobo preučili še orodja OpenNMS in PRTG (angl. *PRTG Network Monitor*).

2.4.1 OpenNMS

OpenNMS [17] je odprtokodno orodje, ki se osredotoča na nadzor storitev, naprav, dogodkov in obvestil. Orodje omogoča nadzor več kot 25 storitev, vključno s HTTP, HTTPS, DHCP in DNS. Vsaki storitvi privzeto pošlje poizvedbo po zmogljivostnih informacijah v 5 minutnih časovnih intervalih. V primeru zaznave odpovedi se frekvenca povpraševanj spremeni. Naslednja funkcionalnost je nadzor omrežnih naprav. OpenNMS povprašuje SNMP agente in prejete podatke hrani s pomočjo orodja RRDTool. RRDTool orodje bomo podrobneje predstavili skupaj z orodjem Cacti. Prejete podatke obdela in jih prikaže v spletnem vmesniku. Pri obsežnejših omrežjih je ročna nastavitev SNMP agentov časovno drago opravilo, zato je potrebno proces v čim večji meri avtomatizirati. Orodje ima definiran poseben identifikator sistema (systemOID), ki se ujema z identifikatorjem naprav. To omogoča pravkar odkritim napravam v omrežju posredovanje nadzorovanih objektov brez posredovanja administratorjev. Tretje področje funkcionalnosti je prejem

obvestil in nadzor dogodkov kot odgovor na prejeta obvestila. Poleg običajnih obvestil ob izpolnitvi pogojev omogoča tudi vključitev SNMP obvestil in PERL skript. OpenNMS je na voljo za večino Linux distribucij, OS X in sistem Windows.

2.4.2 PRTG

PRTG je programska oprema razvijalcev Paessler, ki teče na platformah Windows [18]. Orodje nudi prosto dostopno in komercialno različico. Podatke o omrežju pridobiva s pomočjo protokola SNMP, s prestrezanjem prometa in s storitvijo NetFlow. Cisco je razvil storitev NetFlow za analiziranje prometa IP in nadzorovanje ostalega prometa. Osrednji komponenti NetFlow-a sta zbiralec in analizator. Usmerjevalnik, ki ima omogočeno opcijo NetFlow, generira zapise. Zapisi se nato prenesejo iz usmerjevalnika na zbiralnik, kjer jih analizator obdela in predstavi v prijaznejši obliki. PRTG pokriva široko področje nadzora omrežij z več kot 200 tipi senzorjev. Senzorji pokrivajo nadzor aplikacij, storitev in strojne opreme. Ločijo se po komponentah katere spremljajo, najdemo pa tudi senzorje, ki se razlikujejo le po uporabljenem protokolu za prenos informacij. Uporabnik do stanja in poročil o omrežju dostopa preko spletnega vmesnika ali samostojne aplikacije. Orodje omogoča tudi nadzor preko IPv6, ki vključuje tudi prejem SNMP prometa. Jedro arhitekture orodja je centralizirani strežnik, ki upravlja s sondami na oddaljenih napravah in hrani upravljane podatke. Status, poročila in obvestila so dosegljivi na spletnem strežniku. Centralni strežnik in sonde na oddaljenih napravah tečejo kot Windows storitve. Naslednji element PRTG arhitekture so sonde (angl. *probes*), katerih vloga ustreza obravnavani vlogi agenta v predstavljeni arhitekturi v prejšnjih razdelkih. Sonda predstavlja skupek definiranih senzorjev, ki izvajajo meritve in rezultate samodejno posredujejo strežniku. V primeru odpovedi povezave med centralno točko in oddaljeno sondo se meritve na sondi nadaljujejo in shranjujejo lokalno. Ob vzpostavitvi povezave se podatki posredujejo strežniku. Pri nadzoru oddaljenih omrežij se uporablja varnostni mehanizem SSL. Administrator omrežja pri uporabi pričujočega orodja prejme obvestilo o novo dodani napravi v omrežje. Administrator nato določi katere meritve se bodo na dodani napravi izvajale.

2.4.3 Izbira orodja

V grobem smo spoznali rešitvi PRTG in OpenNMS kot primera komercialnih oz. odprtokodnih rešitev. Podobna orodja na področju nadzorovanja omrežij so še Nagios, MRTG,

SevOne, Solarwinds itd. Različna orodja predstavljajo različne pristope pri izvajanju nadzora omrežnih komponent. Orodja se razlikujejo po uporabljenih protokolih in tehnologijah kot so SNMP, NetFlow, ICMP, MySQL, HTTP in VoIP [19]. Nekatera orodja omogočajo tudi analizo prometa in spremljanje uporabljenih usmerjevalnih protokolov in spremljanje usmerjevalnih tabel. Naslednja točka, v kateri se orodja razlikujejo, so napovedi trendov. Orodje predvidi katera naprava bo dosegla največjo možno obremenitev in s tem uporabniku omogoča izvajanje preventivnih ukrepov. Poleg spremljanja strojne opreme se razlike kažejo tudi pri podpori spremljanja višjenivojskih metrik, kot so odzivnost aplikacij in podatkovnih baz.

3 Orodje Cacti

V prejšnjih razdelkih smo spoznali orodji, katerih funkcionalnosti so poleg nadzora dopolnjevale tudi operacije upravljanja z omrežnimi napravami. V nadaljevanju se bomo posvetili orodju Cacti in njegovi osnovni funkcionalnosti.

3.1 Pregled orodja

Orodje Cacti je odprtokodna rešitev za nadzor omrežnih komponent, ki temelji na MySQL in PHP-ju [20]. Cacti predstavlja nivo uporabniškega vmesnika orodja RRDTool, s pomočjo katerega se podatki hranijo, vizualizirajo in periodično zbirajo.

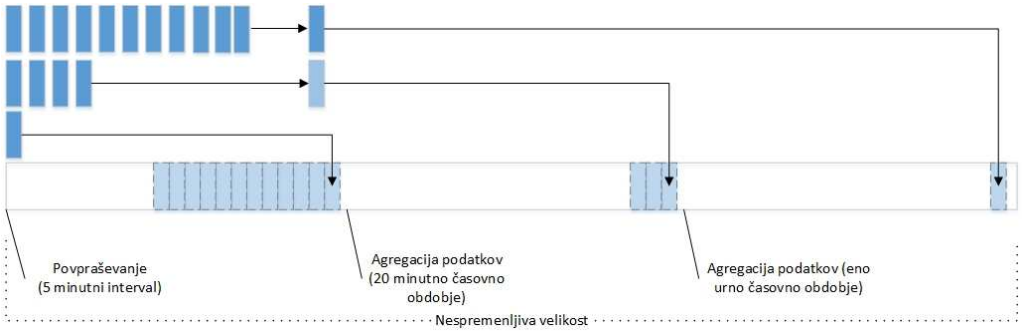
Operacije orodja Cacti delimo na naloge pridobivanja, shranjevanja in predstavitve podatkov. Podatke pridobimo z izvajanjem aplikacije ob konstantnih časovnih intervalih. Privzeto Cacti uporablja PHP skripto *cmd.php* primerno za manjše infrastrukture. Z obsežnejšo infrastrukturo pa se poraja potreba po hitrejšem povpraševanju. V tem primeru je priporočena uporaba aplikacije *Spine*, ki izkorišča večopravilnost modernih operacijskih sistemov in strojne opreme. Za prenos podatkov se uporablja protokol SNMP. Nalogo shranjevanja podatkov vrši orodje RRDTool. RRDTool omogoča hranjenje vrste

podatkov posameznih zaporednih časovnih intervalov z različnih naprav. Pretekle oz. zgodovinske podatke obdeluje s funkcijami kot so povprečje, minimum in maksimum. S takim načinom obdelovanja podatkov ohranjamo najmanjšo zasedenost pomnilniškega prostora. RRDTool ima vgrajena orodja za predstavitev podatkov v grafični obliki. Napisan je v programskem jeziku C in je bil razvit za namene obdelave podatkov posameznih časovnih intervalov, kot so podatki o pasovni širini, obremenjenosti CPE itd. Podatke prejete v nastavljivem časovnem intervalu analizira in rezultate predstavi v grafični obliki. Podatki se shranjujejo v datotekah s končnico `.rdd`. Število zapisov v datoteki se nikoli ne povečuje, ker se stari zapisi konstanto brišejo. V primeru, da podatki niso na voljo, RRDTool shrani zapis v datoteko z vrednostjo *UNKNOWN* [21]. RRDTool bomo podrobneje spoznali v nadaljevanju.

Cacti za svoje nemoteno delovanje potrebuje spletni strežnik, MySQL podatkovno bazo, Net-SNMP in PHP. Razvit je kot spletni vmesnik, zato za delovanje potrebuje spletni strežnik. Za optimalno delovanje je priporočena uporaba strežnikov Apache ali Microsoftova rešitev IIS. MySQL podatkovna baza se uporablja za hranjenje vseh informacij povezanih s prikazom, nastavitvah in podatkov o uporabnikih. Net-SNMP predstavlja programski nabor za uporabo in prilagoditev SNMP protokola. Programski nabor vsebuje ukaznovrstične aplikacije, grafični pregledovalnik MIB objektov, demon za prejem SNMP obvestil in SNMP agenta. Cacti je napisan v programskem jeziku PHP, zato potrebuje predhodno namestitev PHP paketa [3].

3.2 RRDTool

Orodje RRDTool predstavlja hrbtenico orodja Cacti, ki skrbi za shranjevanje in grafično predstavitev podatkov. Prejeti podatki se iz različnih podatkovnih virov shranjujejo v datoteke RRD po metodi FIFO (angl. *first in first out*). Shranjevanje in predstavitev podatkov brez nadaljnje obdelave bi zahtevalo ogromno pomnilniškega prostora. V eni RRD datoteki so definirani različni arhivi imenovani RRA-ji (angl. *Round Robin Archives*), ki omogočajo temelje za agregiranje podatkov. V datoteki arhive sestavlja vsakodnevni, tedenski, mesečni ali celo letni arhiv. Opisani način hranjenja podatkov ne zahteva posebne oblike in vrste podatkov. Podatki, ki jih RRDTool sprejme in shrani, so v obliki izmerjenih vrednosti s pripadajočimi časi meritve. Za dostop in manipulacijo podatkov v `.rrd` datotekah orodje nudi naslednje ukaze:



Slika 3.1 Slika prikazuje metodo shranjevanja v RRD datoteko s tremi definiranimi arhivi [3]. Velikost RRD datoteke je nespremenljiva.

- `create` za generiranje in nastavitev novih RRD datotek,
- `update` za shranjevanje novih vrednosti v datoteko,
- `graph` za predstavitev podatkov v grafični obliki,
- `dump` za izpis vsebine v XML formatu v datoteko ali na standardni izhod,
- `fetch` za zajem podatkov iz navedene datoteke.

Našteli smo le nekaj ukazov, s katerimi se rokuje pri obdelavi podatkov. Primer uporabe ukaza si bomo ogledali tudi pri naslednjem zgledu iz vira [3]. Primer obravnava RRD datoteko s tremi arhivi, ki so prikazani na sliki 3.1. Rezultati povpraševanj s pet minutnim korakom se hranijo v prvem arhivu. Drugi arhiv hrani rezultate agregacije podatkov za 20 minutni interval. Agregirani podatki se hranijo tudi v tretjem arhivu, v katerem podatki predstavljajo obdobje ene ure.

Surovi podatki prispejo vsakih pet minut. Ob prejetju podatkov se osveži celotna RRD datoteka. Po vsaki osvežitvi datoteke se v prvi arhiv zapišejo pravkar prejeti podatki. Po poteku 20 minut se podatkovna množica v prvem arhivu agregira in zapiše v drugi arhiv. Obdelani podatki v drugem arhivu odražajo pogled na časovno obdobje 20 minut. Enak proces se izvede po poteku ene ure. Podatki prvega arhiva se združijo in zapišejo v tretji enourni arhiv, ki nudi širši pogled. Vsak izmed arhivov je omejen s kapaciteto shranjenih podatkov. Ko je ta kapaciteta zasedena, vsak naslednji vnos prepiše najstarejši zapis. Ta način zagotavlja, da se velikost datoteke RRD ne spreminja in ostaja enaka od samega začetka. Slaba stran opisanega načina delovanja je izguba

podrobnih podatkov. Poglejmo si primer še z ustreznimi ukazi orodja, ki ga prikazuje koda 3.1.

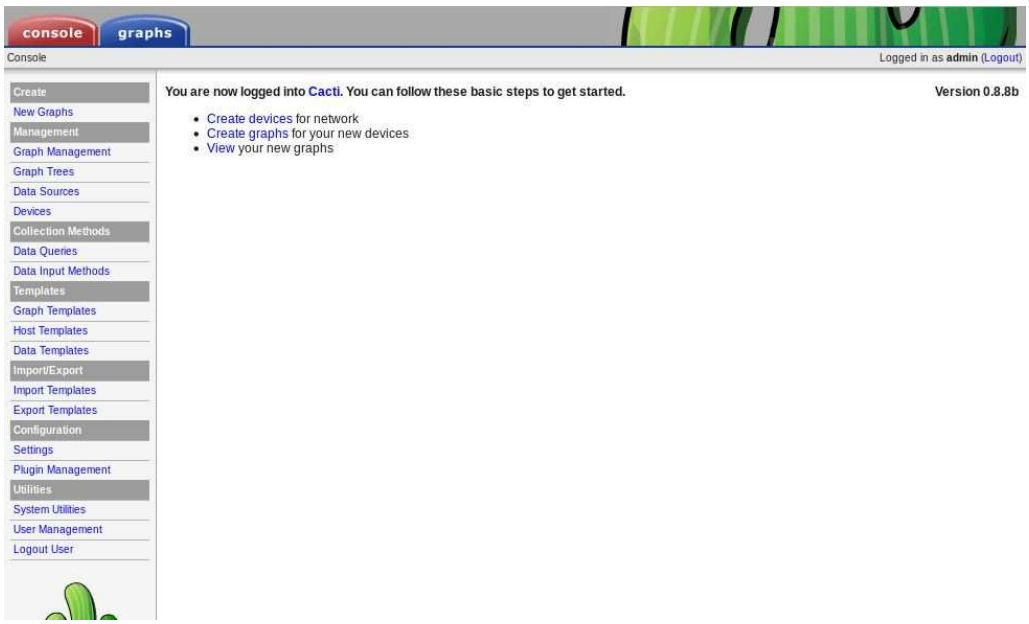
Koda 3.1 Definicija datoteke s podatkovnim virom in tremi arhivi [3].

```
rrdtool create test.rrd -- step 300 \
DS:data:GAUGE:600:U:U                    \
RRA:AVERAGE:0.5:1:16                     \
RRA:AVERAGE:0.5:4:16                     \
RRA:AVERAGE:0.5:12:16
```

Zastavica `--step` določa dolžino intervala v sekundah, s katerim se bodo podatki zapisovali v datoteko. Primer navaja dolžino pet minut, kar je tudi privzeta dolžina. Druga vrstica določa podatkovni vir datoteke imenovan `data`. Podatkovni vir je definiran kot `gauge` podatkovni tip, ki se uporablja za podajanje temperature ali število procesov. Zadnja parametra določata najmanjšo in največjo sprejemljivo vrednost [22]. Za podatkovni vir brez omejitev zapišemo vrednost `U`. Če v 600 sekundah ne prispejo novi podatki, se vrednost postavi na *UNKNOWN*. Za boljše razumevanje si definirajmo pojem podatkovna točka (angl. *data point*). Datoteka v primeru vsebuje tri arhive. Vsak arhiv lahko hrani 16 podatkovnih točk. V prvem arhivu vsaka podatkovna točka odraža pet minutno časovno obdobje. Z vsemi točkami v prvem arhivu pokrivamo 80 minutno obdobje. V drugem arhivu vsaka podatkovna točka predstavlja agregirane podatke zadnjih štirih rezultatov poizvedb. Drugi arhiv lahko tako pokrije največ 320 minut. Z agregacijo posplošujemo in nižamo natančnost podatkov. Podatkovna točka tretjega arhiva vsebuje agregirane podatke zadnjih dvanajstih povpraševanj in nudi pogled na enourno obdobje. Agregacija podatkov nam v tem primeru omogoča hranjenje podatkov tudi do 16 ur nazaj. Ob hranjenju podatkov za več kot zadnjih 80 minut, je potrebna agregacija podatkov, s katero izgubljamo podatkovno granularnost. Privzete definicije arhivov v orodju Cacti so naslednje:

- dnevni (5 minut),
- tedenski (30 minut),
- mesečni (2 uri),
- letni (1 dan).

V oklepaju je navedena raven ločljivosti shranjenih podatkov. V vseh definicijah se podatki pred zapisom povprečijo. Pri orodju Cacti je privzeta dolžina intervala med



Slika 3.2 Do spletnega vmesnika dostopimo z naslovom `http://<streznik>/cacti/`.

povpraševanji pet minut. Najbolj podrobni podatki so vsebovani v dnevnem arhivu. Zaradi velikosti RRD datoteke, se ti podrobni podatki po dveh dneh hranjenja izgubijo. Cacti omogoča nastavitev ločljivosti in s tem čas hranjenja podrobnih podatkov, vendar se velikost datoteke občutno poveča.

3.3 Spletni vmesnik

V podpoglavju 3.1 smo našteli pogoje, ki morajo biti izpolnjeni za uspešno namestitev orodja. Do orodja po namestitvi dostopamo preko spletnega vmesnika na naslovu `http://<streznik>/cacti/`. Pri prvem dostopu se izvedejo še zadnji koraki namestitve. Pri zadnjem koraku potrdimo poti predhodno nameščenih orodij in aplikacij. Poleg poti do orodij RRDTool, PHP in Net-SNMP, nastavimo tudi pot do dnevniške datoteke orodja Cacti. Po končani namestitvi je orodje pripravljeno za uporabo s privzeto aplikacijo za povpraševanje `cmd.php`. Funkcionalnosti in možnosti, ki nam jih nudi Cacti, bomo spoznali preko spletnega vmesnika, ki je prikazan na sliki 3.2. Po vpisu vse operacije in aktivnosti pri upravljanju orodja izvajamo z vlogo administratorja. Začetna stran se imenuje *Console*, do nje pa imajo dostop le administratorji in uporabniki s posebnimi skrbniškimi pravicami.

3.3.1 Zavihek Console

Zavihek *Console* je mesto s katerega upravljamo naše obravnavano orodje. Tu dodajamo nadzorovane naprave, uporabnike in definiramo grafe. Naprava v orodju Cacti predstavlja poleg usmerjevalnikov in računalnikov tudi strežnike. Razdelek Create nudi enostavno definiranje grafov za določeno nadzorovano napravo. Razdelek Management omogoča upravljanje z napravami, grafi, podatkovnimi viri in drevesi grafov. Opise metod in načinov, s katerimi Cacti pridobiva podatke s ciljnih sistemov in naprav, najdemo pod razdelkom Collection Methods. Tukaj rokujemo s podatkovnimi poizvedbami, kot so SNMP metode ali zunanje poizvedbe zunanjih skript. Pod razdelkom Templates najdemo definicije predlog različnih tipov grafov, podatkovnih struktur in končnih uporabnikov. Že obstoječe definicije nam omogočajo enostavno združevanje podatkovnih predlog (angl. *data templates*) v definicije grafov (angl. *graph templates*), ali podatkovne poizvedbe v predloge različnih tipov končnih sistemov (angl. *host templates*). Možnosti uvoza ali izvoza predlog najdemo pod razdelkom uvoz/izvoz (angl. *Import/Export*). Cacti omogoča izvoz ali uvoz tujih predlog zapisanih v razširljivem označevalnem jeziku XML. V okviru razdelka Configuration je moč spremeniti sledeče parametre:

- raven beleženja v dnevniške datoteke,
- različico protokola SNMP,
- lastnosti SNMP protokola (niz skupnosti, število ponovitev poizvedbe, dolžino intervala, v katerem pričakujemo odgovor na poizvedbo),
- število niti in tip aplikacije za poizvedovanje (cmd.php, spine),
- število objektov na SNMP Get zahtevo,
- izvoz in grafične podobe grafov,
- avtentikacijsko metodo (protokol LDAP (angl. *lightweight directory access protocol*), avtentikacija na podlagi spletnega strežnika in lokalno).

Razdelek Utilities nam omogoča dostop in vpogled v dnevniške datoteke ter dostop do sistemskih informacij, kot so število niti, verzija orodja Cacti, sistemske spremenljivke itd.

3.3.2 Zavihek Graphs

Po končani nastavitvi orodja in dodajanju nadzorovanih naprav in sistemov ter definiranju grafov sledi izvajanje nadzora. Končni uporabnik v obravnavanem zavihku skozi množico grafov spremlja aktivnosti in stanje naprav. Vse opazovane naprave v orodju Cacti so organizirane v hierarhična drevesa, s pomočjo katerih se različnim uporabnikom omeji obseg opazovanja. Organiziranje naprav v drevo administratorju omrežij oz. uporabniku orodja Cacti omogoča segmentiranje in logično delitev naprav različnih omrežij in s tem uporabniku omogoča enostaven pregled nad napravami grupiranimi po segmentu omrežja, ali celo fizični lokaciji.

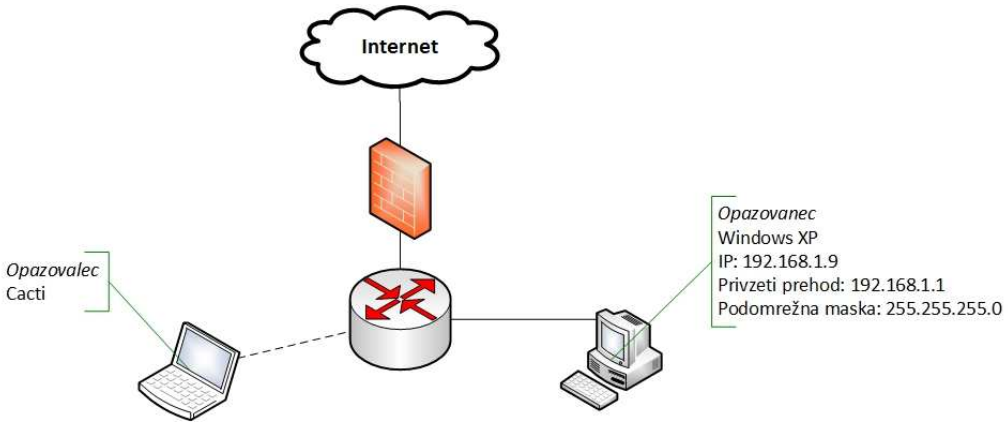
Funkcionalnosti in možnosti orodja smo grobo spoznali skozi spletni vmesnik. Podrobnejšo obravnavo funkcionalnosti bomo spoznali skozi primere uporabe. Pri tem bomo začeli z vzpostavitvijo orodja za nadzorovanje osebnega računalnika v domačem omrežju.

4 Vzpostavitev sistema za nadzor omrežja

V tem poglavju bomo podrobneje spoznali korake pri vzpostavitvi sistema za nadzorovanje v orodju Cacti. Skozi primere bomo odgovorili na vprašanja, kako v orodje dodajamo različne naprave, kako izrisujemo grafe, ter kako dodane naprave organiziramo v hierarhično drevo. Uporabljali bomo zadnjo različico orodja verzije 0.8.8b. Začeli bomo z obravnavo primera domačega omrežja.

4.1 Nadzor osebnega računalnika

V prvem primeru bomo v orodju Cacti dodali napravo, ki se logično in fizično nahaja v lokalnem omrežju. Obe napravi, računalnik na katerem teče pričujoče orodje Cacti in opazovan računalnik, bivata v istem segmentu omrežja in za požarnim zidom, ki nas varuje pred zunanjimi vdori. Na sliki 4.1 sta prikazana računalnika, ki igrata različni vlogi. Na levi strani slike je postavljen sistem z operacijskim sistemom CentOS, na katerem teče Cacti. Na drugi strani leži nadzorovani računalnik na katerem teče operacijski sistem Windows XP z vzpostavljeno storitvijo SNMP. Slednji bo odgovarjal na zahteve s posredovanjem vrednosti zmogljivostnih parametrov. Računalnik, na katerem teče orodje



Slika 4.1 Shema lokalnega omrežja, ki vključuje usmerjevalnik in osebna računalnika. Pri opazovani napravi so zabeleženi osnovni omrežni podatki kot so IP naslov, privzeti prehod in podomrežna maska.

Cacti, je v omrežje povezan brezžično, kar je na sliki prikazano s prekinjeno črto.

4.1.1 Dodajanje naprave

Napravo bomo dodali z uporabo spletnega vmesnika, ki smo ga spoznali v prejšnjem poglavju. V razdelku Management s klikom na povezavo naprave (angl. *Devices*) skočimo na seznam predhodno dodanih naprav. Klik na povezavo *Add* nas pripelje do obrazca za dodajanje naprave. V obrazec vnesemo predstavitveni opis in ime gostitelja ali IP naslov. Na tem koraku napravi določimo tudi predlogo s privzetimi grafi in poizvedbami. Cacti privzeto vsebuje predloge kot so Cisco usmerjevalnik, gostitelj Windows 2000/XP in splošni SNMP gostitelj. Slednje se lahko določi tudi pozneje. Način preverjanja dosegljivosti izberemo v polju *Downed Device Detection*. Izbiramo med orodjem `ping` in SNMP. V našem primeru izberemo slednje. Če ciljni računalnik ne podpira SNMP, se dosegljivost preverja s pingom pri katerem določimo tudi tip protokola in vrata. Zadnji sklop obrazca vsebuje nastavitve SNMP. V obravnavanem primeru bomo uporabljali protokol druge verzije z nizom skupnosti “public” ter UDP vrati 161. Ob uspešni nastavitvi naprave se na vrhu strani prikaže obvestilo, ki je prikazano na sliki 4.2.

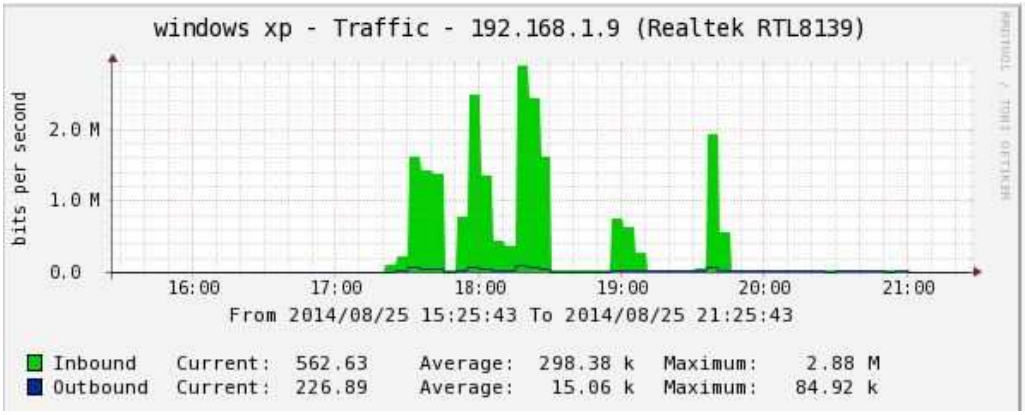
Cacti podatke o zmogljivostnih metrikah prikazuje skozi vrsto grafov. Če v prejšnjih korakih napravi ne določimo predloge gostitelja, preverjamo zgolj njegovo dosegljivost. Pod izbiro predlog gostitelja izberemo gostitelj Windows 2000/XP. S potrditvijo izbire napravi dodamo predhodno definirane predloge grafov in podatkovnih poizvedb. Te predloge vključujejo grafe in poizvedbe o številu prijavljenih uporabnikov v sistem, število

Slika 4.2 Osnovni podatki o napravi, ki jih pridobimo s SNMP poizvedbami.

tekočih procesov, izkoriščenost CPE itd.

4.1.2 Definiranje grafov in organizacija v Cacti drevo

V naslednjem koraku bomo dodali graf za spremljanje aktivnosti na omrežnem vmesniku. Preden bomo napravi pripeli graf, je potrebno dodati ustrezno poizvedbo. Poizvedbo dodamo na istem mestu, kot smo dodali napravo. V razdelku Associated Data Queries so že vsebovane poizvedbe, ki so bile dodane z izborom predloge gostitelja. Za informacije o omrežnih vmesnikih na opazovanem računalniku dodamo poizvedbo *SNMP - Interface Statistics* in sledimo povezavi na vrhu strani Create Graphs for this Host. Na tem mestu so navedeni rezultati poizvedb, med njimi tudi rezultati pravkar dodane poizvedbe. SNMP poizvedba, definirana v interfaces.xml, operira z identifikatorjem objekta 1.3.6.1.2.1.2.2, ki vsebuje vnose o vmesnikih. Pod rezultati slednje poizvedbe najdemo zapis z opisom "Realtek RTL8139 Family PCI Fast Ethernet NIC" ter IP in strojni naslov. Z izbiro vmesnika nato ustvarimo graf, katerega podatki se bodo začeli zajemati z naslednjo ponovitvijo povpraševanj, ki se ponavljajo vsakih pet minut. Tukaj se spomnimo, da je dolžina intervala med povpraševanji nastavljiva. Graf vhodnega in izhodnega prometa na izbranem vmesniku je predstavljen na sliki 4.3. Za definicijo naprave in pripadajočih poizvedb ter grafov sledi umestitev naprave v Cacti drevo. Drevesno strukturo se definira na strani Graph Trees, ki jo najdemo v razdelku Management. V obrazec vnesemo ime drevesa in nato kot elemente drevesa dodajamo poddrevesa, gostitelje ali grafe. Slednji koraki omogočijo uporabniku orodja pregled in nadzorovanje vseh naprav preko enotnega vmesnika v zavihku grafov. Do sedaj smo napravi dodali grafe s pomočjo že definiranih predlog, ki poenostavljajo administrativni postopek. V nadaljevanju bomo spoznali postopek definiranja predloge za grafe in shranjene podatke. Podatkovna predloga ali šablona opisuje podatke, ki jih bo Cacti shranil v RRD datoteke. Predloga temelji na ukazu `create` orodja RRDTool. Definirane predloge zagotavljajo enotnost pri ustvar-

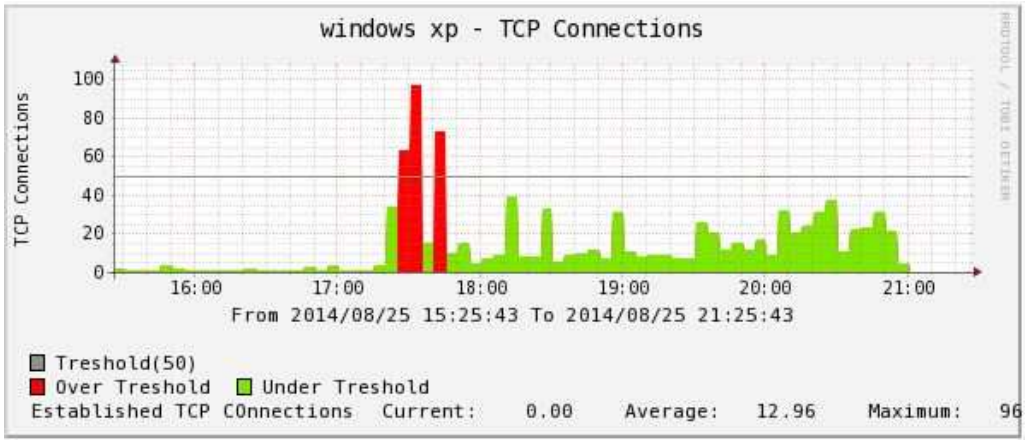


Slika 4.3 Vhodni (zelena) in izhodni (modra) promet v časovnem oknu šestih ur. Na ordinatni osi je označena enota bit/sekundo. Legenda vsebuje tudi trenuten, povprečen in najvišji zabeležen prenos na časovno enoto.

janju RRD datotek. Predloge za graf opisujejo izgled in obnašanje grafa. Definicija v orodju Cacti se prevede v ogrodje funkcije za izris grafa v programu RRDTool s podatkovnimi viri in elementi za prikaz. Sprva bomo definirali podatkovno predlogo za TCP MIB `tcpCurrEstab`. Slednji nam pove število vzpostavljenih TCP povezav. Možnost definicije podatkovne predloge najdemo na strani Data Templates. Poleg imena, rokujemo tudi z metodo poizvedovanja in podatkovnim virom. Pod možnosti podatkovnega vira vnesemo |host_description| - tcpCurrEstab. Niz |host_description| predstavlja eno izmed spremenljivk orodja Cacti, katera se nadomesti z dejanskim opisom gostitelja ob dodelitvi predloge. Podatkovnemu viru je potrebno dodeliti najmanj eden RRA arhiv. Z dodelitvijo različnih tipov arhivov RRDTool ve kako pogosto in koliko časa hrani podatke. V polja vnesemo tudi najmanjšo in največjo pričakovano vrednost s katero omejimo obseg vrnjenih podatkov. Podatki, ki ne ustrezajo predvidenemu intervalu, se v datoteko zapišejo kot neznana vrednost. Vrnjeni podatki imajo določen podatkovni tip, katerega je potrebno tudi definirati. Eden izmed načinov za preverjanje podatkovnega tipa je uporaba ukaza `snmpwalk` v ukazni vrstici, ki razkrije, da je izbrani objekt tipa `Gauge32`. Ker gre za pridobivanje SNMP podatkov, vnesemo tudi identifikator zahtevanega objekta, in sicer 1.3.6.1.2.1.6.9.0. Z vnosom identifikatorja smo končali z definicijo podatkovne predloge. Pridobljene informacije o številu vzpostavljenih TCP povezav na končnem gostitelju bomo prikazali na grafu. Na grafu bomo spremljali število povezav skozi čas, kateri bo ob prekoračitvi definiranih pragov spremenil barvo. Zelena barva bo odražala število TCP povezav v intervalu med 0 do 45, število med 46 in 50 bo označeno z rumeno

barvo. Absoluten prag v pričujočem primeru znaša 50 hkratnih povezav. Vsaka naslednja vzpostavljena povezava botruje k spremembi barve grafa v rdečo. Razlog, da smo izbrali navedene prage, je zgolj predstavitev funkcionalnosti. Tako obnašanje grafa bomo dosegli z uporabo definicij CDEF s katerimi podatkom, ki so prikazani na grafu, dodelimo matematično funkcijo. Območje grafa bomo opremili s pogoji, ki smo jih navedli zgoraj. Gibanje grafa med pogoji bo povzročilo menjavo barve celotnega grafa. CDEF-e dodamo na strani upravljanje z grafi. Posamezno definicijo poimenujemo in ji dodamo element. Elementu v spustnem meniju določimo tip *poljubni niz* ter mu nato vnesemo niz *a,46,50,LIMIT*. CDEF-i se zapisujejo v obliki postfiksnega zapisa. Vpisani niz predstavlja funkcijo, s katero preverjamo ali je vrednost spremenljivke *a* znotraj navedenih mej. Za potrebe našega grafa definiramo še območje v mejah od 0 do 45. Pravkar definirana CDEF-a bomo uporabili pri grafu za spremembo izgleda pri preseganju praga. Predlogo za graf dodamo v razdelku predlog. V polji vnesemo predstavitveno ime in ime, ki je izpisano na grafu. Poleg imen je možno nastaviti višino, širino, samodejno spreminjanje velikosti, itd. Z vnosom imen smo dodali osnovno predlogo na kateri temelji primer. Predlogi dodajmo vodoravno črto s katero označujemo zgornji prag vzpostavljenih povezav. Prag dodamo kot element tipa *HRULE* brez podatkovnega vira. Elementu navedemo še vrednost, barvo in besedilo, ki bo prikazano v legendi. Enake korake ponovimo pri dodajanju elementov s podatkovnim virom *tcpCurrEstb*. Elementom določimo predhodno definirane funkcije CDEF in barve. Končano predlogo lahko sedaj dodelimo napravi. Na izris pravkar dodanega grafa je potrebno počakati na pripadajoče podatke. Prva iteracija povpraševanj ustvari potrebne RRD datoteke. RRDTool za izris grafa potrebuje vsaj dva vnosa v datoteki. Slika 4.4 prikazuje definirani graf za število TCP povezav v zadnjih šestih urah.

V tem podpoglavju smo spoznali kako dodamo končno napravo v Cacti in kako prikažemo stanje naprave. Seznanili smo se tudi z definicijo podatkovnih predlog in postopkom dodajanja poizvedb k dodani napravi. Če želimo zajemati nove podatke, napravi pripnemo podatkovno poizvedbo za želene informacije. Najbolj pogoste poizvedbe so tipa SNMP, ki poizvedujejo po definicijah MIB objektov. Kako definiramo podatkovno predlogo z zajemom podatkov preko SNMP smo spoznali skozi primer. Enak način lahko uporabimo pri definiciji predloge za večino MIB objektov. V okviru tega primera si bomo ogledali še enega izmed načinov kako spremljamo temperaturo na nadzorovani napravi. V Cacti-ju ni vključene poizvedbe za zajem tako podrobnih podatkov. Razlog

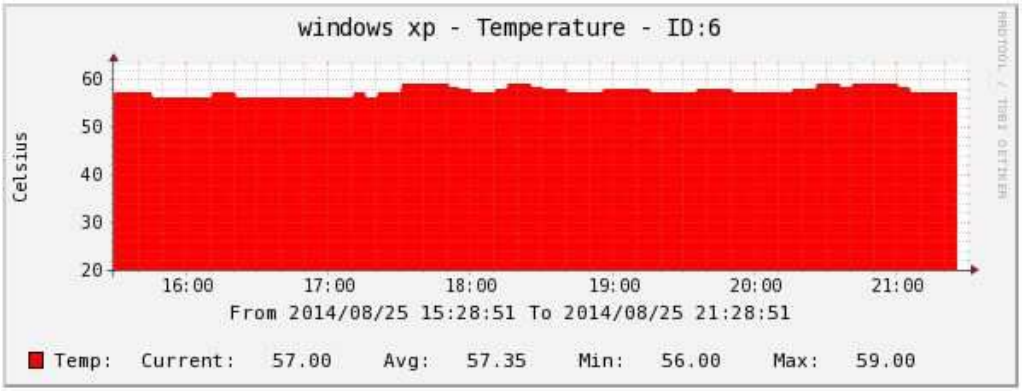


Slika 4.4 Graf je opremljen z legendo in vodoravno črto, ki ponazarja prag. Območje grafa, ki ne preseže praga, je obarvano zeleno. Nad pragom se graf obarva rdeče. Na dnu opazimo število trenutnih, povprečnih in največ doseženih hkratnih povezav.

temu je pomanjkanje oz. odsotnost splošnega MIB objekta. Identifikatorje objektov bomo pridobili s pomočjo programa *SpeedFan* [23]. SpeedFan je program za spremljanje napetosti, hitrosti vrtenja ventilatorjev in temperature. Na opazovani računalnik smo namestili različico 4.50. Nameščeni program informacije z digitalnih senzorjev prikaže samo v uporabniškem vmesniku in jih ne izpostavi v obliki MIB objektov. To dosežemo z namestitvijo SNMP razširitve [24], ki omogoča SNMP dostop do zajetih informacij. Informacije so dostopne preko naslednjih identifikatorjev:

- 1.3.6.1.4.1.30503.1.1.1 - število odčitkov temperature,
- 1.3.6.1.4.1.30503.1.1.2 - število odčitkov hitrosti ventilatorjev,
- 1.3.6.1.4.1.30503.1.1.3 - število odčitkov napetosti,
- 1.3.6.1.4.1.30503.1.2.x - temperature,
- 1.3.6.1.4.1.30503.1.3.x - hitrosti ventilatorjev,
- 1.3.6.1.4.1.30503.1.4.x - napetosti.

Konfiguracija opazovanega računalnika je sedaj končana. Kot smo že spoznali, Cacti ne vsebuje predloge grafa in poizvedbe za zajem naštetih objektov, zato bomo uvozili zunanje. Zahtevane predloge in poizvedbe najdemo na uradnem forumu orodja Cacti [25]. Preden jih uvozimo v Cacti, je potrebno v SNMP poizvedbah speedfan_temp.xml,

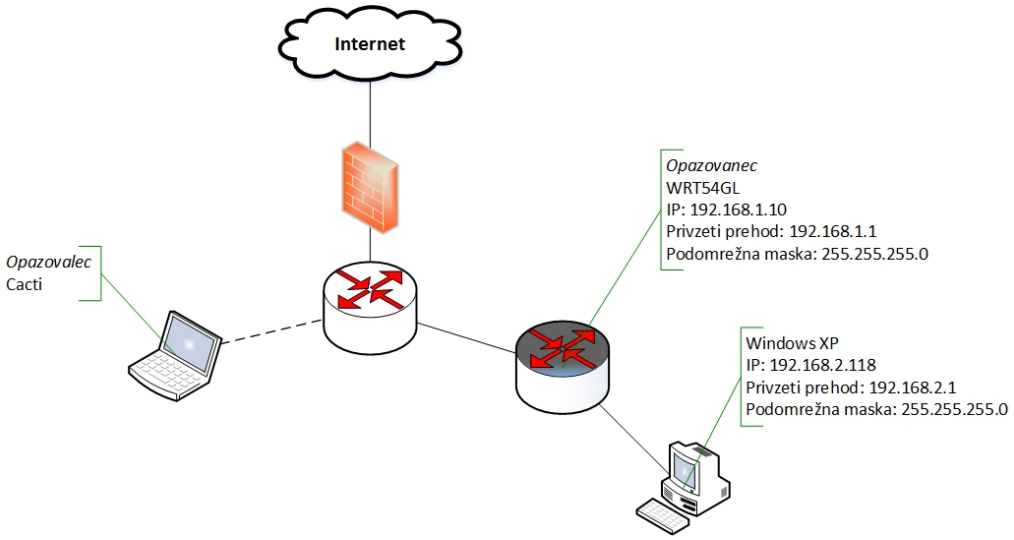


Slika 4.5 Graf spreminjanja temperature na grafični procesni enoti. Temperatura je prikazana v enotah stopinje Celzija.

speedfan_fan.xml in speedfan_volt.xml spremeniti OID-je. To spremenimo v značkah <oid_index> in <oid>. V datoteki speedfan_temp.xml, ki predstavlja poizvedbo za informacije o temperaturah, zapišemo oid 1.3.6.1.4.1.30503.1.2. Skladno z naštetimi identifikatorji popravimo poizvedbo speedfan_fan.xml za hitrosti ventilatorjev in poizvedbo speedfan_volt.xml za zajem informacij o napetosti. Poizvedbe prenesemo v datoteko */cacti/resource/snmp_queries*. Predloge za grafe in podatkovne predloge uvozimo preko spletnega vmesnika. Po vpeljavi vseh poizvedb in predlog nadaljujemo z administracijo naprave v orodju Cacti. Računalniku dodamo podatkovne poizvedbe in prejete rezultate prikažemo z vpeljanimi grafi. Primer grafa prikazuje slika 4.5. V naslovu grafa opazimo niz *ID:6*. Program SpeedFan ne razkriva oz. ne posreduje opisa parametra v MIB objekt, zato se v naslovu grafa uporabi zaporedna številka objekta. S primerjavo rezultatov v ukazni vrstici z ukazom `snmpwalk` in rezultati, ki jih prikazuje aplikacija na opazovanem računalniku, lahko prevedemo zaporedne številke v ustrezne opise. Naslove grafov v orodju Cacti popravimo pod razdelkom Management, kjer niz ID:|query_Index| zamenjamo z ustreznejšim. Ob dodajanju ali zamenjavi strojne opreme je postopek priporočljivo ponoviti, saj tedaj lahko pride do spremembe vrstnega reda. Na opazovanem računalniku mora SpeedFan teči neprekinjeno za izpostavitev in osveževanje podatkov. Orodje SpeedFan nam omogoča pridobiti podrobnejše informacije o nadzorovanem računalniku. Paket *lmSensors* predstavlja podobno rešitev za sisteme Linux.

4.2 Nadzor usmerjevalnika WRT54GL

Naslednji primer, ki si ga bomo ogledali, je nadzor usmerjevalnika. Omrežje iz primera 4.1 smo razširili z usmerjevalnikom Linksys WRT54GL. Opazovani računalnik iz prejšnjega primera smo povezali na dodani usmerjevalnik. Uporabili ga bomo zgolj za generiranje prometa. Vlogo opazovanca igra usmerjevalnik WRT54GL. Slika 4.6 prikazuje omrežje



Slika 4.6 Lokalno omrežje z dodanim usmerjevalnikom WRT54GL. Opazovalec je povezan brezžično, medtem ko je opazovani računalnik prejšnjega primera povezan na WRT54GL.

pričujočega primera. Opazovalec in opazovanec sta povezana v segmentu 192.168.1.0/24. Računalnik z operacijskim sistemom Windows XP je povezan na Ethernet vrata dodanega usmerjevalnika v segmentu 192.168.2.0/24. Pred namestitvijo potrebnih programskih paketov in konfiguracijo v orodju Cacti, si najprej podrobneje oglejmo opazovani usmerjevalnik.

4.2.1 Opis in konfiguracija usmerjevalnika Linksys WRT54GL

Usmerjevalnik s štiri vratnim stikalom omogoča tudi vzpostavitev dostopne točke. Opazovani usmerjevalnik podpira standard 802.11g, ki omogoča dosego 54Mb/s pasovne širine. Standard 802.11g na fizičnem nivoju deluje na frekvenčnem področju 2,4 GHz, ki omogoča združljivost s starejšim standardom 802.11b. Varnost brezžičnega omrežja zagotavlja z uporabo WPA,WPA2 in WEP varnostnimi mehanizmi, ki nudijo različne nivoje varnosti. Najzanesljivejša je uporaba enkripcije z WPA2. Najstarejšega in naj-

ranljivejšega med naštetimi mehanizmi uporabimo samo v primeru združljivosti s starejšimi napravami. Na usmerjevalniku smo omejeni s 16 MB bralno-pisalnega in 4 MB flash bralnega pomnilnika. Podpira namestitev Linux strojne programske opreme (angl. *firmware*), kot je OpenWRT, DD-WRT, TOMATO itd. Na našem usmerjevalniku je nameščen OpenWRT različice *Backfire 10.03.1*. Usmerjevalnik lahko konfiguriramo preko spletnega vmesnika LuCI ali ukazne vrstice. Konfiguracijo preko ukazne vrstice nam omogoča sistem UCI (angl. *Unified Configuration Interface*) [26]. Konfiguraija usmerjevalnika je razdeljena na več datotek, ki se nahajajo v imeniku `/etc/config`. Napravo konfiguriramo preko naslednjih datotek:

- `network` - nastavitve stikala ter logičnega omrežja,
- `system` - osnovne nastavitve celotne naprave, kot je ime gostitelja in časovne zone,
- `firewall` - nastavitev NAT-a in filtriranja paketov,
- `wireless` - nastavitev brezžičnega omrežja.

Z namestitvijo programskih paketov ali storitev, se v imeniku `/etc/config` generirajo pripadajoče konfiguracijske datoteke. Kot v prvem primeru, bomo tudi tukaj izkoristili protokol SNMP za nadzor naprave. Privzeto na usmerjevalniku WRT54GL ni nameščene storitve, ki bi odgovarjala na zahteve SNMP. Pri namestitvi izbiramo med demonom *mini-snmpd* ali *snmpd*. V našem primeru bomo namestili slednjega. Demon *mini-snmpd* je okrnjenja implementacija SNMP demona, ki je namenjen za uporabo v vgrajenih sistemih (angl. *embedded systems*) z omejenimi pomnilniškimi viri. Ta uporablja le 32-bitne števce, kar v definiciji MIB objekta predstavlja podatkovni tip Counter32. Okrnjenost zasledimo tudi v konfiguracijski datoteki, v kateri nastavljamo le osnovne atribute, kot so niz skupnosti, lokacija, kontakt in seznam priklopnih točk datotečnega sistema (angl. *filesystems mountpoints*) ter omrežnih vmesnikov. Če ima usmerjevalnik dostop do interneta, lahko snmpd demon namestimo z ukazom `opkg install snmpd`.

Vse nadaljnje konfiguracije nameščenega demona izvedemo v datoteki `/etc/config/snmpd`. V konfiguracijski datoteki je poleg že naštetih osnovnih parametrov, mogoče konfigurirati tudi SNMP agenta. Preden pogledamo v samo vsebino konfiguracijskih datotek, spoznajmo njihovo osnovno sintakso. Zaporedje `rezervirana_beseda parameter vrednost` prikazuje elemente sintakse. Z rezervirano besedo `config` definiramo začetek odseka tipa `parameter` z imenom `vrednost`. Znotraj odseka z besedo `option` lastnosti

`parameter` priredimo vrednost `vrednost`. Za prireditev večjega števila vrednosti lastnosti, uporabimo rezervirano besedo `list` [26]. Obravnavo konfiguracijske datoteke demona snmpd začnimo z nastavitvijo vrat na katerih posluša agent. To dosežemo z nastavitvijo številke vrat parametru `agentaddress`. Datoteka vključuje konfiguracijo nadzora dostopa, ki ga agent podpira. Podpora nadzora dostopa je razlog, da v konfiguracijski datoteki prepozna izraze, kot so `com2sec`, `group`, `view` in `access`. Navedeni izrazi nastopajo v vlogi parametra. Odsek tipa `com2sec` definira preslikavo para niza skupnosti in opisa izvora, ki je lahko podan z imenom gostitelja, podomrežjem ali besedo `default` v podano ime varnosti (angl. *security name*). Opis izvora z besedo `default` omogoča dostop vsem, ki poznajo niz skupnosti. Pri dohodnih paketih se izbere prva ujemajoča kombinacija. Tip odseka `group` navaja preslikavo modela varnosti in imena varnosti v skupino. V katero skupino pripada preslikava para podamo z imenom skupine. Pod model varnosti podamo različico protokola SNMP. Znotraj odseka model varnosti podamo v parametru `version`, ki lahko vsebuje verzijo `v1`, `v2c` ali `usm`. Slednji predstavlja varnost oz. avtentikacijo na podlagi uporabniških imen in je privzeti način pri tretji verziji protokola. Ime varnosti podamo v parametru `secname`. Vrednost slednjega parametra mora ustrezati imenu varnosti, ki smo ga podali v odseku `com2sec`. V konfiguracijski datoteki definiciji imen varnosti in skupin sledita definiciji pravic za dostop in pogleda. Pogled definiramo v odseku tipa `view`. Pogledu poleg imena podamo tudi tip in poddrevo identifikatorjev OID. Tip pogleda določimo v parametru `type` in zasede vrednost bodisi `included` bodisi `excluded`. S podanim tipom določimo ali je poddrevo parametra `oid` zajeto v pogled ali ne. Zadnji sklop konfiguracije SNMP agenta vključuje nastavitev pravic dostopa definirani skupini ali modelu varnosti. Pravice podamo v odseku `access`. V odseku zopet podamo verzijo protokola SNMP. Omejenosti na določeno verzijo se izognemo s prireditvijo vrednosti `any`. Najnižjo raven avtentikacije in enkripcije vhodnih zahtev določimo s parametrom `level`. Raven brez avtentikacije dosežemo z vpisom vrednosti `noauth`. Enkripcijo dosežemo z vrednostjo `priv`. Znotraj odseka agentu priredimo tudi ozadje oz. kontekst in pogoje pri ujemanju konteksta s prihajajočimi zahtevami. Kontekst srečamo pri uporabi SNMP protokola tretje verzije. Predstavlja skupino nadzorovanih informacij do katerih dostopajo agenti. Določena nadzorovana informacija lahko nastopa v različnih kontekstih. Ravno tako ima lahko agent dostop do množice kontekstov. V našem primeru bi z definiranjem konteksta podali dodaten pogoj pri preverjanju pravic in obsegu dostopa do nadzorovanih informacij. Zato pri

parametru `context` zapišemo `none`. Ujemanje predpone ali celotnega konteksta vhodnih podatkovnih enot določimo v parametru `prefix`. Parametrom `read`, `write` in `notify` določimo ime pogleda definiranega v odseku pogleda. S slednjimi navedbami določimo do katerih poddreves objektov imamo bralni ali pisalni dostop. Določimo pa tudi katere vrednosti objektov se lahko pošlje kot obvestilo [27]. Del konfiguracije demona *snmpd* je predstavljen v kodi 4.1.

Koda 4.1 Izvorna koda konfiguracijske datoteke demona *snmpd* na usmerjevalniku WRT54GL.

```
config agent
        option agentaddress UDP:161
config com2sec public
        option secname ro
        option source default
        option community secret
config group public_v1
        option group public
        option version v1
        option secname ro
config view all
        option viewname all
        option type included
        option oid .1
config access public_access
        option group public
        option context none
        option version any
        option level noauth
        option prefix exact
        option read all
        option write none
        option notify none
config system
        option sysLocation      'home'
        option sysContact       'wrt@example.com'
        option sysName          'Linksys'
```

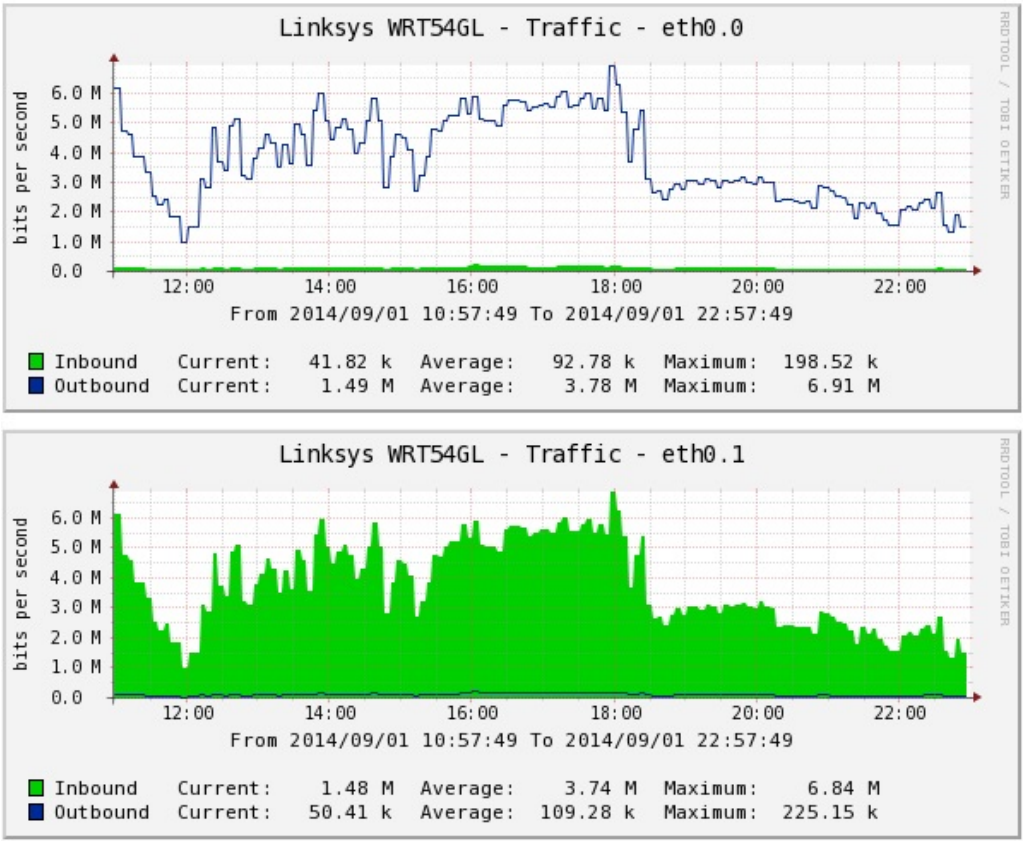
Kot je navedeno v konfiguraciji, agent posluša na vratih 161. Privzeto je promet SNMP blokiran s strani požarnega zida. Če sledimo konfiguraciji našega omrežja, je potrebno v nastavitvah požarnega zida dodati le pravilo, ki dovoli prejem zahtev naslovljenih na vrata 161. Z zapisom pravila v datoteko `/etc/config/firewall` je konfiguracija na strani usmerjevalnika WRT54GL končana. Povezljivost in delovanje lahko preverimo z že poznanim ukazom `snmpwalk`. Ukaz izvedemo na strani opazovalca z naslovom 192.168.1.10 in nizom skupnosti "secret". Niz skupnosti smo določili v nastavitvah

agenta. Niz je naveden v izvorni kodi 4.1 v odseku `com2sec`.

4.2.2 Primeri meritev

Usmerjevalnik v orodje Cacti dodamo na nam že poznani način, ki smo ga spoznali pri dodajanju osebnega računalnika v prejšnjem primeru 4.1. Naslov IP, številka vrat in niz skupnosti so vsi podatki, ki jih potrebujemo pri dodajanju naprave. V konfiguraciji 4.1 agenta smo na opazovancu v odseku `group` navedli prvo verzijo protokola SNMP. V orodju Cacti verzijo protokola ter ostale možnosti konfiguriramo skladno z nastavitvami usmerjevalnika. Pri izbiri predloge gostitelja izberemo usmerjevalnik Cisco. Napravi nato priredimo poizvedbe in predloge grafov. Konfiguracija na strani opazovalca je končana z umestitvijo naprave v drevo Cacti. S poizvedbo "SNMP - Interface Statistics" pridobimo podatke o sedmih vmesnikih. V zajete rezultate poizvedbe je vključena tudi povratna zanka (angl. *loopback*) tipa `softwareLoopback`. Prejete informacije o vmesnikih ne opisujejo le fizičnih vmesnikov usmerjevalnika. Poleg povratne zanke so v odgovoru tudi zapisi o vmesnikih `eth0`, `eth0.0`, `eth0.1`, `br-lan`, `wl0` in `wl0.1`. Eno izmed fizičnih komponent usmerjevalnika predstavlja štiri vratno stikalo, ki je programabilno. To stikalo je povezano na vmesnik, ki mu ustreza identifikator `eth0`. Identifikator `wl0` se nanaša na prisotno strojno opremo radijskega dela v sistemu. Opisi vmesnikov `eth0.0`, `eth0.1` in `wl0.1` ustrezajo definiranim navideznim vmesnikom v pripadajočih konfiguracijskih datotekah. Vmesnik mosta, ki vse navidezne in fizične vmesnike logično poenoti v en omrežni vmesnik, je opisan z nizom `br-lan`.

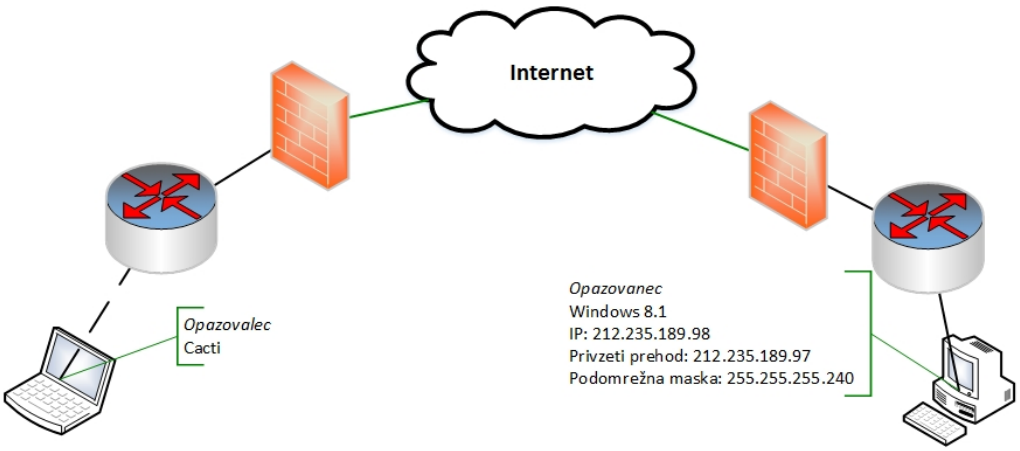
Slika 4.7 prikazuje grafa statistike prometa. Prevedimo opazovana vmesnika v fizična vrata usmerjevalnika. Po konfiguraciji usmerjevalnika pripadajo vrata stikala z oznakami 1-4 vmesniku `eth0.0`. Vrata WAN pripadajo preostalemu vmesniku. Na sliki opazimo znatno podobnost med grafom izhodnega prometa na vmesniku `eth0.0` in grafom vhodnega prometa na vmesniku `eth0.1`. Zabeleženo aktivnost prenosa podatkov je generiral in sprejemal osebni računalnik, ki je bil povezan na vrata z oznako 2. Izhodni promet vmesnika `eth0.0` v večji meri predstavlja vhodni promet vmesnika `eth0.1`. Vhodni in izhodni promet na slednjem vmesniku vključuje tudi SNMP zahteve in odgovore. Z natančnejšo analizo in ravnijo zajetih podatkov je razlika statistike prometa na vmesnikih opaznejša, saj so zahteve SNMP naslovljene le na usmerjevalnik in niso posredovane na osebni računalnik.



Slika 4.7 Statistika prometa na navideznih vmesnikih `eth0.0` in `eth0.1`. Grafa pokrivata enako časovno okno opazovanja usmerjevalnika. Vhodni (zelena) in izhodni (modra) promet so podani v prenesenih bitih na sekundo. Grafa sta opremljena z legendo in najmanjšo, največjo in povprečno količino prenesenih podatkov.

4.3 Nadzor oddaljenega računalnika

V zadnjem primeru se bomo lotili vzpostavitve nadzora oddaljenega računalnika. Računalnik v Laboratoriju za računalniške strukture in sisteme na Fakulteti za računalništvo in informatiko bomo opazovali iz domačega omrežja na lokaciji Novo mesto. V primerih 4.1 in 4.2 sta bila opazovanca v istem omrežju kot opazovalec. Pri pošiljanju zahtev po upravljavskih podatkih smo uporabili bodisi verzijo 1 ali verzijo 2c. Varnost obeh verzij temelji na nizu skupnosti, ki se preko omrežja pošlje kot golo besedilo. Če povezave ne zaščitimo z drugimi načini, si lahko prisluškovalec prometa prilasti niz skupnosti. Napadalec lahko s prestreženim nizom skupnosti poleg zbiranja informacij o sistemu tudi spreminja nastavitve upravljanega sistema. Shema omrežja pričujočega primera je prikazana na sliki 4.8. Poizvedbe generirane v orodju Cacti se bodo pošiljala preko prostranega omrežja na IP naslov 212.235.189.98. V tem primeru bomo za komunikacijo med opazovancem in



Slika 4.8 Opazovani računalnik je prikazan na desni strani slike. Opazovalec in opazovanec se nahajata na različnih fizičnih lokacijah med katerima poteka komunikacija s protokolom SNMP verzije 3.

opazovanim računalnikom uporabili SNMP protokol verzije 3, ki zagotavlja avtentikacijo in zasebnost med sodelujočima entitetama.

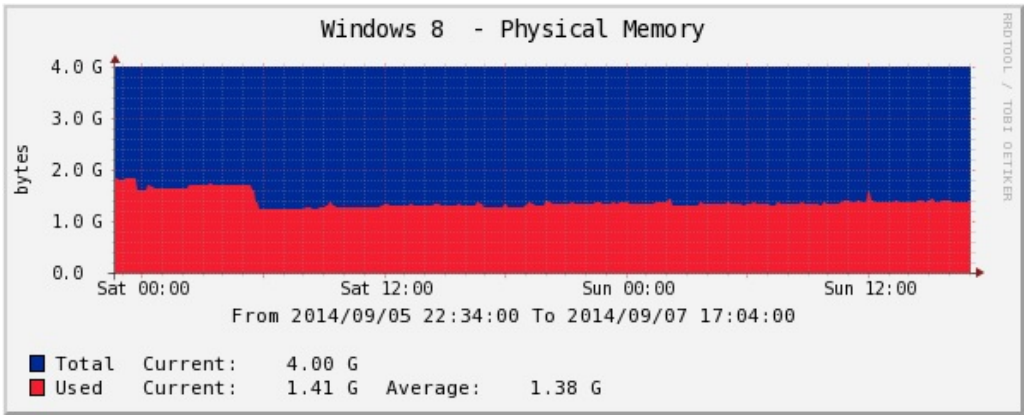
4.3.1 Konfiguracija opazovanca

Na oddaljenem računalniku teče operacijski sistem Windows, ki inherentno ne podpira protokola SNMPv3. Zato bomo Microsoftovo rešitev nadomestili s SNMP agentom MG-SOFT [28]. Nameščeni agent s podporo SNMPv3 nudi nadzor in upravljanje omrežij z MD5 ali SHA1 zgoščevalnim algoritmom ter enkripcijo vsebine paketov SNMP z DES

ali AES blokovnim algoritmom. Oddaljenega agenta konfiguriramo z uporabniškim imenom in geslom uporabnika, v imenu katerega se preverja istovetnost identitete in izvaja šifriranje sporočila. Definiranemu uporabniku omejimo obseg opazovanih parametrov z navedbo konteksta in naborom SNMP operacij. Agentu nastavimo tudi kriptografsko zgoščevalno funkcijo, ki se uporabi pri tvorjenju kode za avtentikacijo sporočila. Pri izračunu avtentikacijske kode se uporabi vsebina sporočila s tajnim ključem, ki je poznan prejemniku in pošiljatelju. Ob prejemu sporočila entiteta uporabi predhodno konfigurirani tajni ključ za ponovni izračun kode za overitev sporočila. Ujemanje prejete in izračunane kode predstavlja prejemniku dokaz, da je pošiljatelj pooblaščeni upravljavec. Zaščito pred napadom s ponovitvijo SNMPv3 rešuje s časovnim oknom, v katerem omejimo čas prejema sporočila. Funkcionalnost je dosežena s preverjanjem časovnega okna in sinhronizacijo. Entiteti, ki komunicirata, hranita dve vrednosti, in sicer število ponovnih zagonov komponente SNMP ter število sekund od zadnjega ponovnega zagona. Sinhronizacija med pošiljateljem in prejemnikom se doseže z vključitvijo navedenih vrednosti v glavo sporočila. Prejemnik par vrednosti shrani lokalno za vsakega izmed upravljavcev. S prejemom nadaljnjih sporočil od upravljavca se lokalni vrednosti posodobita. Prejemnik sporočila ali upravljana entiteta šibko sinhronizacijo z upravljavcem ohranja s povečevanjem lokalno shranjenega časa od zadnjega ponovnega zagona upravljavčeve SNMP entitete. Upravljana entiteta lokalni vrednosti zapiše v sporočilo, ki je poslano upravljavcu. Sporočilo je pravočasno, če sta prejeti vrednosti znotraj upravljavčevih meja. Konfiguracijo agenta končamo z izbiro algoritma za enkripcijo SNMP podatkov. Polega algoritma podamo tudi tajni ključ, ki se uporabi pri procesu šifriranja vsebine sporočila.

4.3.2 Konfiguracija opazovalca

Oddaljeni računalnik s poznanim IP naslovom in številko vrat dodamo v Cacti na nam že poznani način. Nastavitev se razlikuje le v uporabljeni verziji protokola SNMP. Definiciji naprave vnesemo varnostne parametre ter ustrezne avtentikacijske in enkripcijske algoritme, ki se ujemajo z nastavitvami oddaljenega agenta. Kljub temu, da na oddaljenem računalniku teče operacijski sistem Windows 8, ga v orodju Cacti lahko opredelimo s predlogo gostitelja Windows 2000/XP. V zadnjem koraku vnesemo še podatkovne poizvedbe in predloge grafov za grafični prikaz opazovanih zmogljivostnih parametrov. Oddaljenemu računalniku se v petminutnih intervalih pošiljajo poizvedbe po informaci-



Slika 4.9 Graf predstavlja izkoriščenost glavnega pomnilnika. Od 4 GB razpoložljivega fizičnega pomnilnika, ki je na grafu prikazan z modro barvo, je v povprečju uporabljenih 1,38 GB.

jah o zasedenosti posamezne diskovne particije, izkoriščenosti jeder CPE, izkoriščenosti fizičnega pomnilnika, številu tekočih procesov in aktivnosti na omrežnih vmesnikih. Graf izkoriščenosti razpoložljivega fizičnega pomnilnika je prikazan na sliki 4.9.

Z uporabo zadnje verzije protokola SNMP smo vpeljali avtentikacijo in zaupnost podatkov s šifriranjem podatkovne enote protokola. Prvotno verzija protokola ni bila podprta, zato smo posegli po rešitvi neodvisnega razvijalca. Podporo vseh treh verzij SNMP dosežemo tudi z uporabo programskega paketa Net-SNMP.

5 Zaključek

V diplomskem delu smo se lotili vzpostavitve sistema za nadzor omrežnih naprav s programski orodjem Cacti. Preden smo se lotili analize orodja smo spoznali metode, s katerimi spremljamo različne metrike na opazovanih napravah. Pri nadzoru dosegljivosti nam zadostujejo že preprost ukaz, kot je naprimer `ping`. Če želimo nadzorovati kompleksnejše parametre kot so izgube paketov na določenih omrežnih vmesnikih ali število vzpostavljenih TCP povezav, pa moramo poseči po naprednejših orodjih, kot je obravnavano orodje Cacti. Uporaba takih orodij zahteva temeljitejše razumevanje infrastrukture, ki jo z orodjem vzpostavimo. Sodelujočim entitetam v infrastrukturi razdelimo dve vlogi. Vlogo upravljavca igra centralni strežnik ali sistem, s katerega se pošilja zahteve upravljani napravi. Na upravljani napravi teče agent, ki stanje naprave izpostavi v množici nadzorovanih objektov različnih podatkovnih tipov. Vsakega izmed upravljanih objektov je mogoče identificirati z zaporedjem imen ali števil. Prenos nadzorovanih podatkov med upravljavcem in opazovano napravo omogoča protokol SNMP. Obravnavo smo nadaljevali z analizo metod in načinov, s katerimi Cacti podatke obdela in shranjuje. Cacti stanje opazovanih naprav predstavlja z množico grafov. Podatki, ki so grafično predstavljeni,

se črpajo iz datotek orodja RRDTool. V zaključni fazi pričujočega dela smo z obravnavanim orodjem spremljali lokalne in oddaljene naprave. Pri konfiguraciji upravljavca in naprav smo se srečali z različnimi metodami prenosa podatkov med entitetama. Najpogostejša metoda zajemanja podatkov orodja Cacti je pošiljanje SNMP zahtev, vendar orodje podpira zajem podatkov tudi z orodji ukazne vrstice, kot je PERL skripta. Orodje se je izkazalo kot zelo prilagodljivo, saj omogoča definiranje ali uvoz lastnih predlog grafov ali gostiteljev. Ravno tako pri uporabi nismo bili omejeni s tipom ali operacijski sistem, ki teče na opazovani napravi.

Protokol SNMP poleg načina delovanja zahteva-odgovor, omogoča tudi pošiljanje obvestil. Cacti temelji na povpraševanju, torej Cacti v konstantnih časovnih intervalih pošilja zahteve vsem opazovanim napravam. Orodje prejme le podatke, ki jih je zahteval v poslanih sporočilih. Pošiljanje obvestil prestavlja asinhroni dogodek, v katerem opazovana naprava brez predhodne zahteve pošlje obvestilo navedenemu prejemniku. Prejete zahteve je mogoče prevesti v zapis dnevniške datoteke. V nadaljnjem delu bi se osredotočili na razvoj lastne rešitve, s katero bi omogočili prejem SNMP obvestil preko obdelave dnevniških datotek. Rešitev bi razvili kot spletni vmesnik, v katerem bi uporabnik prejel obvestilo o prejemu SNMP obvestila.

Literatura

[1] J. Kurose, K. Ross, Computer Networking: A Top-Down Approach, Pearson Education, Limited, 2010.

[2] A. Deveriya, Network Administrators Survival Guide, Vol. 10401, Pearson Education, 2005.

[3] T. Urban, Cacti 0.8 Beginner's Guide, Packt Publishing Ltd, 2011.

[4] E. Rosen, Vulnerabilities of Network Control Protocols: An example, RFC 789 (Jul. 1981).
url: http://www.ietf.org/rfc/rfc789.txt

[5] ping(8) - Linux man page, http://linux.die.net/man/8/ping, [Online; accessed 29-August-2014].

[6] K. Papagiannaki, R. Cruz, C. Diot, Network Performance Monitoring at small time scales, in: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ACM, 2003, pp. 295–300.

[7] J. Case, R. Mundy, D. Partain, B. Stewart, Introduction and Applicability Statements for Internet-Standard Management Framework, RFC 3410 (Informational) (Dec. 2002).
url: http://www.ietf.org/rfc/rfc3410.txt

[8] K. McCloghrie, M. Rose, Management Information Base for Network Management of TCP/IP-based internets, RFC 1156 (Historic) (May 1990).
url: http://www.ietf.org/rfc/rfc1156.txt

[9] K. McCloghrie, D. Perkins, J. Schoenwaelder, Structure of Management Information Version 2 (SMIv2), RFC 2578 (INTERNET STANDARD) (Apr. 1999).
url: http://www.ietf.org/rfc/rfc2578.txt

[10] S. Routhier, Management Information Base for the Internet Protocol (IP), RFC 4293 (Proposed Standard) (Apr. 2006).
url: http://www.ietf.org/rfc/rfc4293.txt

[11] R. Raghunarayan, Management Information Base for the Transmission Control Protocol (TCP), RFC 4022 (Proposed Standard) (Mar. 2005).
url: http://www.ietf.org/rfc/rfc4022.txt

[12] B. Fenner, J. Flick, Management Information Base for the User Datagram Protocol (UDP), RFC 4113 (Proposed Standard) (Jun. 2005).
url: http://www.ietf.org/rfc/rfc4113.txt

[13] D. Mauro, K. Schmidt, Essential SNMP, "O'Reilly Media, Inc.", 2005.

[14] R. Presuhn, Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), RFC 3416 (INTERNET STANDARD) (Dec. 2002).
url: http://www.ietf.org/rfc/rfc3416.txt

[15] R. Rivest, The MD5 Message-Digest Algorithm, RFC 1321 (Informational), updated by RFC 6151 (Apr. 1992).
url: http://www.ietf.org/rfc/rfc1321.txt

[16] F. PUB, Secure hash standard (shs).

[17] T. Balog, Enterprise-Wide Network Management with OpenNMS, http://www.oreillynet.com/pub/a/sysadmin/2005/09/08/opennms.html?page=2, [Online; accessed 8-August-2014].

[18] Paessler, PRTG Network Monitor, http://www.paessler.com/manuals/prtg8, [Online; accessed 9-August-2014].

[19] A. Leskiw, Product Comparison: SevOne 5.3 vs. Solarwinds Network Performance Monitor 10.5, http://www.networkmanagementsoftware.com/sevone-solarwinds-compared, [Online; accessed 11-August-2014].

[20] I. The Cacti Group, What is Cacti, http://www.cacti.net/what_is_cacti.php, [Online; accessed 12-August-2014].

[21] D. Kundu, S. I. Lavlu, Cacti 0.8 Network Monitoring, Packt Publishing Ltd, 2009.

[22] T. Oetiker, RRDTool Documentation, http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html, [Online; accessed 19-August-2014].

[23] A. M. Comparetti, SpeedFan, http://www.almico.com/speedfan.php, [Online; accessed 21-August-2014].

[24] A. Gembe, SpeedFan SNMP Extension, http://deve.loping.net/projects/sfsnmp/, [Online; accessed 21-August-2014].

[25] Speedfan Script and Templates , http://forums.cacti.net/viewtopic.php?f=12&t=23476, [Online; accessed 21-August-2014].

[26] The UCI System, http://wiki.openwrt.org/doc/uci, [Online; accessed 1-September-2014].

[27] Configuration Files, http://docs.oracle.com/cd/E19624-01/817-2559-16/config.html, [Online; accessed 1-September-2014].

[28] MG-SOFT SNMP Master Agent, http://www.mg-soft.com/agent.html, [Online; accessed 5-September-2014].